



Universidad de Almería

Proyecto Fin de Carrera

*“DESARROLLO DE UNA APLICACION
WEB PARA LA GESTION Y SEGUIMIENTO
EVOLUTIVO DE PACIENTES A PARTIR DE
MEDIDAS NEUROFISIOLOGICAS. UN CASO
DE ESTUDIO CON POTENCIALES
EVOCADOS COGNITIVOS”*

INGENIERÍA INFORMÁTICA

Alumno: Emilio López Plaza

Director: Francisco Gabriel Guil Reyes

Almería, Septiembre 2011

Agradecimientos

Agradezco y dedico este proyecto a mi familia, y en especial a mis padres, por su comprensión, motivación y el apoyo que siempre me han brindado para lograr todas y cada una de las metas que me he propuesto. Sin ellos todo esto no sería posible.

A mi director de proyecto, Francisco Guil Reyes, por ser el guía, consejero y ayudante de todo este trabajo. Porque cada vez que lo he necesitado ha estado ahí y ha sido mi principal apoyo durante la realización de este proyecto. Del mismo modo quiero agradecer a Mayte Daza su inestimable ayuda en cuanto a temas de los que yo desconocía y que sin su contribución me hubiesen resultado mucho más difíciles de comprender.

A mi novia y amigos, por ayudarme a olvidar los malos momentos y hacerme disfrutar de los buenos. Porque sin ellos me hubiese resultado infinitamente más difícil llegar hasta aquí.

A todos ellos, y a los que faltan (que son muchos), ¡MIL GRACIAS!

Índice general

1. Introducción	13
1.1. Motivación.	13
1.2. Objetivos.	15
1.3. Metodología.	15
1.4. Planificación temporal.	17
1.5. Recursos.	17
1.5.1. Hardware.	17
1.5.2. Software.	19
1.6. Organización de la memoria.	20
1.7. Bibliografía básica.	20
1.7.1. Bibliografía del dominio.	20
1.7.2. Bibliografía de implementación.	21
1.7.3. Bibliografía de metodología.	21
2. Potenciales Evocados	23
2.1. Introducción a los Potenciales Evocados.	23
2.1.1. Definición.	24
2.2. Procedimiento de obtención de Potenciales Evocados.	24
2.2.1. El problema del ruido.	25
2.3. Características.	28
2.4. Clasificación.	31
2.4.1. Potenciales Evocados Visuales.	31
2.4.2. Potenciales Evocados Auditivos.	32
2.4.3. Potenciales Evocados Somatosensoriales.	33
2.5. El componente P300.	34
2.5.1. Obtención de la respuesta.	34
2.5.2. Parámetros del P300.	36
2.5.3. Aplicaciones.	36
3. Especificación de <i>AWARD Potentials</i>	39
3.1. Plan de trabajo.	39
3.2. Funcionalidad del sistema y Casos de uso.	41
3.2.1. Actores del diagrama de casos de uso.	42
3.2.2. Casos de uso.	42
3.3. Diagrama de clases informal.	47

4. Diseño e Implementación	49
4.1. Clases de diseño.	49
4.2. Diagramas de Secuencia.	52
4.3. Diseño de la Interfaz de usuario.	62
4.4. Diseño de la base de datos.	63
4.4.1. Base de datos de <i>AWARD Potentials</i>	63
4.4.2. Unificación con la base de datos de <i>AWARD Subject</i>	68
5. Aspectos destacados de Implementación	71
5.1. Diseño de aplicaciones web en PHP.	71
5.1.1. ¿PHP o ASP?	72
5.1.2. Conexión con la Base de Datos MySQL.	73
5.1.3. Inserción, eliminación y modificación de registros.	74
5.1.4. Scripts de autenticación y sesiones.	76
5.2. La librería JSXGraph.	79
5.2.1. Estudio previo de la metodología usada.	79
5.2.2. ¿Por qué no Java o Flash?	79
5.2.2.1. ¿Por qué JavaScript y SVG en lugar de Java/Flash? . . .	80
5.2.2.2. ¿Como representar gráficos en un navegador sin usar Java o Flash?	81
5.2.3. JSXGraph	82
5.3. Hojas de estilo CSS.	83
5.3.1. Estructura de la hoja de estilos.	83
5.3.2. Tipos de elementos CSS implementados.	84
5.3.2.1. Clases.	85
5.3.2.2. Identificadores.	85
5.3.2.3. Pseudoclasas.	86
5.3.2.4. Pseudoelementos.	86
5.3.3. Plantillas CSS utilizadas.	86
5.3.4. Especificaciones de CSS.	87
5.4. Funciones del editor gráfico en JavaScript.	88
5.4.1. Función insertar etiqueta.	88
5.4.2. Función insertar area.	89
5.4.3. Función insertar distancia.	91
5.4.4. Función mostrar diferencia.	93
5.5. Aplicación web rica usando AJAX.	95
5.5.1. El objeto <i>XMLHttpRequest</i>	97
5.5.2. Uso de la tecnología AJAX en <i>AWARD Potentials</i>	99
6. Evaluación de <i>AWARD Potentials</i>.	101
6.1. Validación.	101
6.2. Verificación.	102
6.3. Aceptación de los usuarios.	103
6.3.1. Cuestionario de validación: explicación y funcionalidad.	103
6.3.2. Cuestionario sobre oportunidad y previsiones futuras	104
6.3.3. Críticas, mejoras, sugerencias y errores destacados.	104

7. Conclusiones y trabajos futuros.	105
7.1. Conclusiones.	105
7.2. Trabajos futuros propuestos.	106
A. Manual de instalación y usuario de <i>AWARD^{Potentials}</i>	109
A.1. Instalación de software.	109
A.1.1. Instalación de software previo.	109
A.1.2. Instalación de <i>AWARD^{Potentials}</i>	113
A.2. Manual de usuario.	115
A.2.1. Sesión del terapeuta.	115
A.2.2. Gestión de sujetos.	118
A.2.3. Gestión de gráficos (potenciales evocados).	123
B. Manual de usuario de <i>MaxG EP-01</i>	129
B.1. Aspecto inicial.	129
B.2. Controles de memorias.	131
B.2.1. Elección de memoria de promediado.	131
B.3. Realización de tests.	132
B.3.1. Crear un test nuevo.	132
B.3.2. Actualizar un test ya creado.	133
B.3.3. Borrar un test.	133
Bibliografía	134

Índice de figuras

1.1. Captura del Software	14
1.2. Test Potenciales	14
1.3. Recursos hardware para Potenciales Evocados Visuales	19
1.4. Recursos hardware para Potenciales Evocados Auditivos	19
2.1. Sistema Internacional 10-20	24
2.2. Detección de Campo Magnético mediante sensores SQUID	25
2.3. Señal contaminada por ruido	26
2.4. Ejemplo de promediación de ondas para la obtención del PEP	27
2.5. Índice de correlación CI vs SNR	28
2.6. Polaridad de componentes. Positivos (P) y negativos (N)	29
2.7. Distribución de los sensores por el cuero cabelludo	30
2.8. Test de Potencial Evocado Visual en bebés	31
2.9. Test de Potencial Evocado Auditivo en bebés	32
2.10. Test de Potencial Evocado Somatosensorial en adultos	33
2.11. Representación en los ejes cartesianos de los componentes P3a y P3b	35
2.12. Paradigma Odd-ball	35
2.13. Parámetros del P300	36
3.1. Diagrama de Gantt (planificación temporal)	40
3.2. Diagrama de Casos de Uso completo	41
3.3. Diagrama de Casos de Uso “Solicitar cuenta”	42
3.4. Diagrama de Casos de Uso “Administrar nuevos terapeutas”	42
3.5. Diagrama de Casos de Uso “Manejar sesiones”	43
3.6. Diagrama de Casos de Uso “Administrar sujetos”	44
3.7. Diagrama de Casos de Uso “Administrar gráficos”	45
3.8. Diagrama de Casos de Uso “Editar graficos”	46
3.9. Diagrama de Clases (Fase de análisis)	48
4.1. Diagrama de Clases de la fase de Diseño	50
4.2. Clase ConexionBD	51
4.3. Clase Funciones	51
4.4. Clase Grafico	52
4.5. Clases Login y Logout	52
4.6. Diagrama de Secuencia “Iniciar Sesión”	53
4.7. Diagrama de Secuencia “Terminar Sesión”	54
4.8. Diagrama de Secuencia “Eliminar Sujeto”	55

4.9. Diagrama de Secuencia “Registrar Nuevo Sujeto”	56
4.10. Diagrama de Secuencia “Actualizar Sujeto”	57
4.11. Diagrama de Secuencia “Combinar Gráficos”	58
4.12. Diagrama de Secuencia “Cargar Gráfico”	59
4.13. Diagrama de Secuencia “Editar Gráfico Simple”	61
4.14. Interfaz de usuario de <i>AWARD Potentials</i>	62
4.15. Base de Datos de <i>AWARD Subject</i>	69
5.1. Bitmap vs SVG	80
5.2. Renderización SVG o VML	81
5.3. JSXGraph en dispositivo móvil	83
5.4. Primera plantilla CSS	84
5.5. Segunda plantilla CSS	84
5.6. Función insertarEtiqueta()	89
5.7. Función insertarArea()	91
5.8. Función insertarDistancia()	93
5.9. Función mostrarDiferencia()	94
5.10. Tecnologías agrupadas bajo el concepto AJAX.	95
5.11. Comparación modelo tradicional y AJAX.	96
5.12. Comparación entre comunicaciones síncronas y asíncronas.	97
5.13. Funcionamiento del objeto <i>XMLHttpRequest</i>	98
5.14. Ejemplo de script validación AJAX.	100
7.1. Unificación con la base de datos de <i>AWARD Subjects</i>	107
A.1. Sitio web de XAMPP	110
A.2. Descarga de XAMPP para Windows	110
A.3. Ventana de instalación de XAMPP	111
A.4. Panel de control de XAMPP	111
A.5. Página de estado de XAMPP	112
A.6. Página de seguridad de XAMPP	113
A.7. Página de seguridad de XAMPP, cambio de contraseña	114
A.8. Importación de base de datos SQL desde PHPMyAdmin	114
A.9. Archivos fuente de <i>AWARD Potentials</i>	115
A.10. Captura de pantalla index.php	116
A.11. Captura de pantalla terapeuta.php	116
A.12. Captura de pantalla buscar_sujetos.php	117
A.13. Captura de pantalla nuevo_sujeto.php	118
A.14. Captura de pantalla sujetos.php	119
A.15. Captura de pantalla perfil_sujeto.php	120
A.16. Captura de pantalla eliminar_sujeto.php	120
A.17. Captura de pantalla editar_sujeto.php	121
A.18. Captura de pantalla subir_imagen.php	122
A.19. Captura de pantalla subir_imagen.php (2)	122
A.20. Captura de pantalla capturar_imagen.php	123
A.21. Captura de pantalla graficos.php	124
A.22. Captura de pantalla grafico.php	125

A.23. Captura de pantalla editar_grafico.php	126
A.24. Captura de pantalla editar_grafico_multiple.php	127
B.1. Aspecto del software MaxG EP-01	129
B.2. Aspecto inicial de MaxG EP-01	130
B.3. Condiciones de adquisición	132

Índice de tablas

1.1. Planificación temporal de tareas.	18
4.1. Tabla Terapeuta de la Base de datos.	63
4.2. Tabla Sujeto de la Base de datos.	64
4.3. Tabla Gráfico de la Base de datos.	65
4.4. Tabla <i>Attention Network Test</i> de la Base de datos.	66
4.5. Tabla <i>Tarea Stroop</i> de la Base de datos.	66
4.6. Tabla <i>Tarea Sternberg</i> de la Base de datos.	67
4.7. Tabla <i>Tarea 2-back</i> de la Base de datos.	67
4.8. Tabla <i>Tarea Go-noGo</i> de la Base de datos.	68
5.1. Comparativa JavaScript, Java y Flash.	80
6.1. Cuestionario de validación.	103
6.2. Cuestionario sobre oportunidad y previsiones futuras.	104
6.3. Cuestionario de críticas, mejoras y sugerencias.	104

1

Introducción

1.1 Motivación.

Uno de los principales objetos de estudio de la Psicología Cognitiva es la cognición humana y, para abordarlo, esta disciplina ha utilizado distintas técnicas. Una de las más utilizadas es el registro de potenciales evocados (PE). La utilización de esta técnica se está extendiendo cada vez más en distintos campos y ámbitos de aplicación con son: atención, procesamiento de la información, memoria, comprensión del lenguaje y procesos emocionales, entre otros. En concreto, uno de los trastornos en el que los potenciales evocados nos proporcionan información útil, debido a su alta relación con el proceso de atención, es el llamado Trastorno por Déficit de Atención/Hiperactividad. [1]

Los estudios de potenciales evocados se utilizan para medir la actividad eléctrica del cerebro en respuesta a estímulos visuales, auditivos o somatosensoriales. Los estímulos que se envían al cerebro a través de cada uno de estos sentidos provocan minúsculas señales eléctricas. Estas señales se trasladan por los nervios y la médula espinal hasta regiones específicas del cerebro; los electrodos las registran, las amplifican y las muestran para que el terapeuta las interprete. En la Figura 1.1 se puede ver un ejemplo de los potenciales evocados resultantes de un test realizado con estimulación visual.

Los estudios de potenciales evocados suelen utilizarse de manera intensiva por los profesionales de la salud en un amplia gama de casos. Además de ayudar al diagnóstico y seguimiento evolutivo del TDAH, se utilizan para evaluar la audición o la vista, en especial en bebés y niños, para diagnosticar trastornos del nervio óptico y para detectar tumores u otros problemas que afectan al cerebro y la médula espinal. También pueden utilizarse para evaluar el funcionamiento del cerebro durante un coma. Los exámenes que detectan anomalías en las funciones sensoriales, por lo general no arrojan un diagnóstico específico de qué es lo que causa la anomalía, aunque son bastante fiables a la hora de detectarlas.

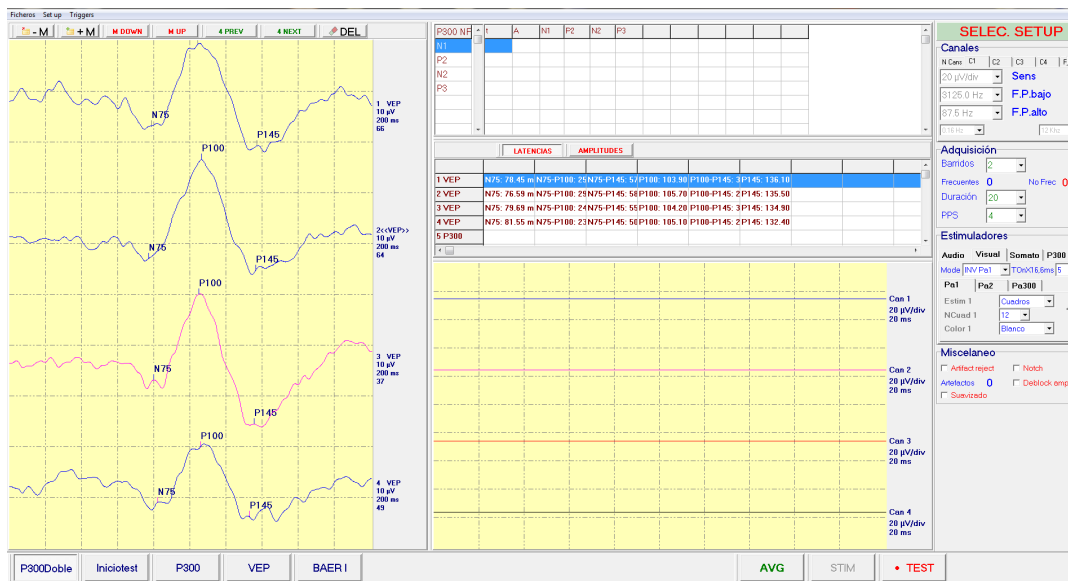


Figura 1.1: Potenciales Evocados Visuales (Captura del Software).

Cabe destacar el uso del componente (u onda) “P300”, el cual será objetivo principal de nuestro proyecto. Se trata de una onda con un patrón establecido que suele aparecer (aproximadamente a los 300 ms) ante estímulos inesperados que son relevantes para la tarea que ha de realizar el sujeto. Su amplitud se ve modulada por el grado de expectación que el sujeto tiene ante un determinado estímulo. El P300 está considerado como un índice de la memoria de trabajo, en la que la amplitud nos informa acerca de la asignación de los recursos atencionales y su latencia es un indicador del tiempo de evaluación del estímulo. Según la literatura, suele estar asociado con el proceso de atención, y suele recibir el nombre de Potencial Cognitivo.

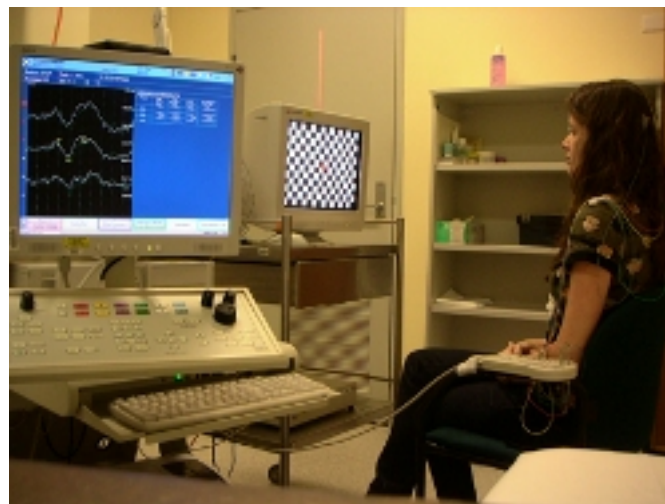


Figura 1.2: Ejemplo de test con Potenciales Evocados Visuales.

En la Figura 1.2 se muestra un ejemplo de sesión para la captura de señales neurofisiológicas visuales. Como se puede apreciar, el input visual es un tablero de ajedrez,

que alterna los colores de forma periódica. En el caso de la P300, el test sería algo más complejo, mostrando un input no previsto (p.e. pantalla roja) de forma aleatoria y con una frecuencia bastante menor a la del input base (suele estar en torno a un 25%). Este input no frecuente es el que produce el potencial P300 como respuesta neurofisiológica al estímulo inesperado.

1.2 Objetivos.

El objetivo del proyecto consiste en el diseño e implementación de una herramienta basada en web para la gestión y seguimiento evolutivo de sujetos (pacientes) con alteraciones en el Sistema Nervioso Central, a la que hemos llamado *AWARD^{Potentials}*. Esta herramienta se alimentará de datos contextuales y medidas neurofisiológicas obtenidas a partir de un hardware específico diseñado para la obtención de potenciales evocados.

Como ejemplo de población diana, se propone un caso de estudio centrado en sujetos que presentan el conocido Trastorno por Déficit de Atención/Hiperactividad (TDAH). Este tipo de trastorno es una patología neuroconductual común en la infancia que provoca en el sujeto alteraciones atencionales, impulsividad y sobreactividad motora; sintomatología que va a tener importantes repercusiones tanto en su desarrollo como en su capacidad de aprendizaje y ajuste social. Desde la aproximación de la Neurociencia Cognitiva, se ha estudiado su relación con el funcionamiento de la Red de Control Ejecutivo o Atención Ejecutiva, existiendo la posibilidad de planificar sesiones de entrenamiento para la intervención y mejora cognitiva-conductual de esta población (*Daza et al., 2010*). Mediante la realización programada de tareas neurocognitivas se ha demostrado que los sujetos experimentan una mejora cualitativa en cuanto a capacidad de atención. Sin embargo, esta mejora sólo es perceptible desde un punto de vista comportamental. Con la idea de evaluar la eficacia de este tipo de entrenamiento con otro tipo de medidas objetivas, como pueden ser las de tipo neurofisiológico (potenciales evocados), se plantea la realización de una herramienta para la edición y análisis de señales neurofisiológicas como mecanismo para el seguimiento evolutivo de estos sujetos.

Esta herramienta se programará íntegramente en PHP y JavaScript con la intención de que cualquier terapeuta interesado en este estudio tenga total accesibilidad a la aplicación. La base de datos se realizará en MySQL y el tratamiento de los gráficos de potenciales y su interacción con la base de datos se realizará íntegramente de forma asíncrona mediante el uso de la tecnología AJAX para proporcionar una experiencia web rica y más agradable al terapeuta.

1.3 Metodología.

Para el desarrollo del sistema aquí propuesto de manera efectiva, se ha procedido a descomponerlo en una serie de actividades. A su vez, cada actividad consta de una serie de procesos y técnicas. Con este procedimiento, se puede realizar el proyecto de una forma mucho más estructurada.

A continuación se procede a listar cada una de estas actividades y procesos de forma jerárquica.

1. **Tarea:** Estudio de la viabilidad del proyecto.
 - a) Documentación preliminar sobre el problema del TDAH.
 - b) Búsqueda de información sobre los avances tecnológicos en el campo de los Potenciales Evocados Cognitivos.
 - c) Investigación sobre los proyectos existentes centrados en el tratamiento de Potenciales Evocados.
 - d) Especificación formal de la idea.
2. **Tarea:** Recopilación de la información disponible del hardware asignado al proyecto y familiarización con el mismo.
 - a) Documentación del hardware.
 - b) Prueba de los distintos tipos de captación de potenciales.
 - 1) Captura de Potenciales Evocados Visuales.
 - 2) Captura de Potenciales Evocados Auditivos.
 - c) Prueba de captura del componente P300.
 - d) Recopilación y estudio de la información obtenida.
3. **Tarea:** Implementación de la Base de Datos de *AWARD^{Potentials}*.
 - a) Estudio de las bases de datos utilizadas en otros Proyectos *AWARD*.
 - b) Implementación de la base de datos de gestión de usuarios de la aplicación.
 - c) Implementación de la base de datos de los tests de Potenciales Evocados.
4. **Tarea:** Diseño e implementación del entorno de usuario.
 - a) Implementación del entorno del terapeuta.
 - 1) Creación de la biblioteca básica de funciones.
 - 2) Implementación de la gestión de usuarios (búsqueda e inserción de nuevos sujetos) y su interacción con la base de datos (creación de consultas SQL).
 - 3) Implementación de la información estadística ofrecida a los terapeutas.
 - b) Implementación del entorno del sujeto.
 - 1) Creación de la biblioteca básica de funciones.
 - 2) Implementación de la gestión de usuarios (modificación e inserción de sujetos) y su interacción con la base de datos (creación de consultas SQL).
 - 3) Implementación del sistema de gestión de tests.
 - c) Implementación del resto de entornos comunes.
5. **Tarea:** Análisis e implementación del visualizador de tests.

- a) Estudio de la librería JSXGraph [2].
 - b) Estudio de las necesidades de los usuarios y funcionalidades a implementar.
 - c) Implementación del sistema de transferencia de datos desde el hardware a la aplicación y su incorporación a la base de datos.
 - d) Implementación del visualizador de tests.
 - e) Implementación del visualizador de tests simple.
 - 1) Creación del entorno de visualización.
 - 2) Implementación del editor de etiquetas
 - 3) Implementación del calculador de áreas.
 - 4) Implementación del selector de contenidos.
 - f) Implementación del visualizador de tests comparativo.
 - 1) Creación del entorno de visualización comparativo.
 - 2) Creación del área de múltiples test.
 - 3) Implementación del medidor de distancias.
 - 4) Implementación del diferenciador de tests.
 - 5) Implementación del selector de tests.
6. **Tarea:** Integración de los diferentes módulos del sistema.
7. **Tarea:** Pruebas y verificación de las funcionalidades del sistema.
8. **Tarea:** Elaboración de la memoria del proyecto.

1.4 Planificación temporal.

Una vez conocidas las diferentes tareas que requiere nuestro sistema, podemos desarrollar la planificación temporal del mismo, tal y como se muestra en la Tabla 1.1.

1.5 Recursos.

1.5.1. Hardware.

Los recursos hardware que se han utilizado en la realización del proyecto son los siguientes:

1. Módulo **EP-01** de *MaxG Sistemas*. Dispositivo hardware diseñado para la captura de Potenciales Evocados.
2. Módulo **AuStim**, estimulador auditivo de *MaxG Sistemas*, diseñado para la generación de estímulos auditivos.
3. Módulo **ViStim**, estimulador visual de *MaxG Sistemas*, diseñado para la generación de estímulos visuales.

Tarea	Actividades	Tiempo
1. Estudio de la viabilidad del proyecto	A. Documentación preliminar sobre el problema del TDAH	5 días
	B. Búsqueda de información sobre los avances tecnológicos en el campo de los Potenciales Evocados Cognitivos	2 días
	C. Investigación sobre los proyectos existentes centrados en el tratamiento de Potenciales Evocados	3 día
	D. Especificación formal de la idea	3 días
2. Recopilación de la información disponible del hardware asignado al proyecto y familiarización	A. Documentación del hardware	2 días
	B. Prueba de los distintos tipos de captación de potenciales	
	1) Captura de Potenciales Evocados Visuales	1 día
	2) Captura de Potenciales Evocados Auditivos	1 día
C. Prueba de captación del componente P300	2 días	
D. Recopilación y estudio de la información obtenida	4 días	
3. Implementación de la Base de Datos de <i>AWARD^{Potentials}</i>	A. Estudio de las bases de datos utilizadas en otros Proyectos <i>AWARD</i>	2 días
	B. Implementación de la base de datos de gestión de usuarios	1 día
	C. Implementación de la base de datos de los tests de P.E.	2 días
4. Diseño e implementación del entorno de usuario	A. Implementación del entorno del terapeuta	
	1) Creación de la biblioteca básica de funciones.	5 días
	2) Implementación de la gestión de usuarios (búsqueda e inserción de nuevos sujetos) y su interacción con la base de datos (creación de consultas SQL).	7 días
	3) Implementación de la información ofrecida a los terapeutas	5 días
	B. Implementación del entorno del sujeto	
	1) Creación de la biblioteca básica de funciones.	4 días
	2) Implementación de la gestión de usuarios (modificación e inserción de sujetos) y su interacción con la base de datos (creación de consultas SQL)	8 días
3) Implementación del sistema de gestión de tests	12 días	
C. Implementación del resto de entornos comunes	5 días	
5. Análisis e implementación del visualizador de tests	A. Estudio de la librería JSXGraph	4 días
	B. Estudio de las necesidades de los usuarios y funcionalidades a implementar	2 días
	C. Implementación del sistema de transferencia de datos desde el hardware a la aplicación y su incorporación a la base de datos	2 días
	D. Implementación del visualizador de tests simple	
	1) Creación del entorno de visualización	4 días
	2) Implementación del editor de etiquetas	3 días
	3) Implementación del calculador de áreas	5 días
	4) Implementación del selector de contenidos	1 día
	E. Implementación del visualizador de tests comparativo	
	1) Creación del entorno de visualización comparativo	1 día
	2) Creación del área de múltiples test	2 días
	3) Implementación del medidor de distancias	3 días
	4) Implementación del diferenciador de tests	5 días
	5) Implementación del selector de tests	2 días
	6. Integración de los diferentes módulos del sistema	5 días
7. Pruebas y verificación de las funcionalidades del sistema	7 días	
8. Elaboración de la memoria del proyecto	47 días	
TOTAL	167 días	

Tabla 1.1: Planificación temporal de tareas.

4. Auriculares Panasonic RP-HT225.
5. Electrodo.
6. WebCam Conceptronic.
7. Ordenador HP Intel Core i3 3.07GHz 64 bits, 4Gb RAM.



Figura 1.3: Recursos hardware para Potenciales Evocados Visuales. (I) Módulo ViStim. (II) Módulo EP-01. (III) Electrodo.



Figura 1.4: Recursos hardware para Potenciales Evocados Auditivos. (I) Módulo AuStim. (II) Auriculares.

1.5.2. Software.

Los recursos software que se han utilizado en la realización del proyecto han sido los siguientes:

1. Software de desarrollo **Eclipse PHP Development Tools Project**.
2. Entorno de captura de señales (potenciales evocados) **MaxG EP-01**.
3. Librería de javascript **JSXGraph**.

4. Base de datos **MySQL** 5.5.8.
5. Entorno **PHP** 5.3.5.
6. Servidor **Apache** 2.2.17.

1.6 Organización de la memoria.

La memoria del proyecto está organizada de la siguiente forma:

- En el Capítulo 2, **Introducción a los Potenciales Evocados**, se hará un breve resumen de lo que es un potencial evocado así como sus principales características y los casos de estudio concretos de interés para este proyecto. Todo esto se estudiará enfocando el contenido al estudio del tratamiento del TDAH.
- En el Capítulo 3, **Especificación de *AWARD^{Potentials}***, se describirá el plan de trabajo seguido, así como el resultado de la fase inicial de análisis.
- En el Capítulo 4, **Diseño e implementación**, se desarrollará el diseño de todos los módulos de la aplicación. Este capítulo albergará todos los diagramas de clases y de secuencia, además de información relativa al diseño de la base de datos de *AWARD^{Potentials}*.
- En el Capítulo 5, **Aspectos destacados de implementación**, se explicarán los detalles de implementación más importantes.
- En el Capítulo 6, **Evaluación de *AWARD^{Potentials}***, se tratará tanto la prueba como el proceso de implantación de la herramienta, así como la especificación del formulario diseñado para medir el grado de aceptación de los usuarios.
- En el Capítulo 7, **Conclusiones y trabajos futuros**, se explicarán las conclusiones y los trabajos futuros que se derivan de este proyecto.

1.7 Bibliografía básica.

Podemos denominar bibliografía básica aquella que ha resultado más útil para la realización del proyecto. Esta bibliografía se puede dividir en tres apartados: (1) la relativa al dominio de la aplicación, (2) la relativa a la implementación de la herramienta y, por último, (3) aquella relacionada con la metodología de desarrollo del software.

1.7.1. Bibliografía del dominio.

Esta bibliografía está íntimamente relacionada con el dominio de la aplicación. Con esto nos referimos a los recursos bibliográficos de estudios sobre Potenciales Evocados o el TDAH. Como casos concretos podemos citar los textos de:

- [1] Núñez-Peña M. I. *“Potenciales evocados cerebrales en el contexto de la investigación psicológica: una actualización”*.

- [3] Michael D. Rugg and Michael G. H. Coles “*Electrophysiology of mind: event-related brain potentials and cognition*”.

1.7.2. Bibliografía de implementación.

Las principales referencias bibliográficas a la hora de desarrollar la herramienta han sido las usadas para el desarrollo del sitio web en PHP y su conexión con la base de datos así como las utilizadas para el aprendizaje del lenguaje javascript y su modalidad asíncrona usando XML y JSON (AJAX) y por último las utilizadas como referencia básica para el uso de la librería de manejo de gráficos en javascript, JSXGraph.

- [4] Becerra, A. and Torres, M. “*Desarrollo de aplicaciones PHP para bases de datos*”
- [5] Goodman D. and Morrison M. and Novitski P. and Gustaff Rayl C. “*JavaScript Bible, 7th Edition*”.
- [6] Torres, M. and Martínez, S. “*Introducción al desarrollo de aplicaciones web ricas para bases de datos usando Ajax*”
- [2] JSXGraph Reference Guide.

1.7.3. Bibliografía de metodología.

La bibliografía relacionada con la metodología de desarrollo de la herramienta y sus extensiones, así como aquellas obras genéricas o básicas para la realización de cualquier proyecto software.

- [7] Roger S., Pressman “*Ingeniería del Software. Un enfoque práctico*”.
- [8] I., Sommerville “*Ingeniería del Software. 6th Edition*”.

2

Potenciales Evocados

2.1 Introducción a los Potenciales Evocados.

Como ya se ha comentado anteriormente, uno de los principales objetos de estudio de la Psicología es la cognición humana y, para abordarlo, esta disciplina ha utilizado distintas técnicas. En este proyecto se presentará una técnica concreta: el registro de potenciales evocados (PE). Esta técnica ha permitido ampliar el conocimiento sobre algunas operaciones cognitivas a partir del comportamiento de ciertos potenciales eléctricos cerebrales [1]. El argumento que subyace a la utilización de esta técnica de registro para investigar estos procesos psicológicos es el siguiente: dado que los procesos cognitivos se suponen relacionados con la actividad eléctrica cerebral, y puesto que los PE son fluctuaciones de los potenciales eléctricos del cerebro provocadas por la ocurrencia de un suceso (o por la presentación de un estímulo), parece que estos últimos pueden ser útiles para ayudarnos a comprender mejor el sistema cognitivo humano. En definitiva, se considera que los PE son indicadores de procesos (o subprocesos) cognitivos o perceptivos. Aspectos, entre otros, como la latencia del proceso cognitivo, su presencia/ausencia o su relación con otros procesos, podrán determinarse a través de la información que obtengamos mediante este tipo de técnica de registro. De hecho, basta con echar una ojeada a la literatura sobre PE para comprobar que su utilización se está extendiendo cada vez más en varios campos de la psicología cognitiva: atención, procesamiento de la información, memoria, comprensión del lenguaje y procesos emocionales, entre otros.

A continuación, se expondrá brevemente qué es un PE; cual es el procedimiento que hay que seguir para obtenerlo; cuáles son sus características definitorias; cómo suelen clasificarse; y, finalmente, qué información nos aportan, es decir, cuál es su utilidad.

2.1.1. Definición.

Los PE son fluctuaciones en el voltaje del electroencefalograma (EEG) provocadas por sucesos sensoriales, motores o cognitivos. Se ha observado que tras la presentación de un determinado tipo de estímulo o la realización de alguna tarea concreta, se producen cambios constantes en el EEG. Estos cambios adoptan la forma de picos (o valles) y se supone que nos informan acerca de los procesos cognitivos subyacentes [1].

Por lo que se refiere a la nomenclatura que usualmente se utiliza para referirse a un PE concreto, encontramos que la terminología es variada. En ocasiones se utilizan como sinónimos los términos pico, onda o componente. Algunos autores, sin embargo, han tratado de ser más precisos y han definido el término componente como algo distinto de un pico o una onda. Definen componente como una porción del registro de la actividad eléctrica cerebral, sensible a una manipulación experimental concreta, que se considera un reflejo de un proceso particular o de un grupo de procesos. En este sentido, un componente puede ser tanto un único pico como una secuencia de picos. Ocurre, sin embargo, que en muchas ocasiones todavía se siguen utilizando los términos pico y onda indistintamente como si fueran sinónimos de componente [3].

2.2 Procedimiento de obtención de Potenciales Evocados.

Los PE se obtienen colocando electrodos sobre distintas posiciones en la cabeza, para lo que se utilizan usualmente las propuestas en el Sistema Internacional 10-20 (véase la Figura 2.1). En este sistema, cada localización viene definida por dos coordenadas: una, su proximidad a una región concreta del cerebro (frontal, central, temporal, parietal u occipital); y, otra, su ubicación en el plano lateral (números impares para la izquierda, números pares para la derecha y la letra z para las localizaciones centrales). Aparte de estas localizaciones, en algunos experimentos se colocan electrodos sobre otras zonas del cuero cabelludo (por ejemplo sobre el área de Wernicke o sobre el área de Broca, ambas relacionadas con el lenguaje).

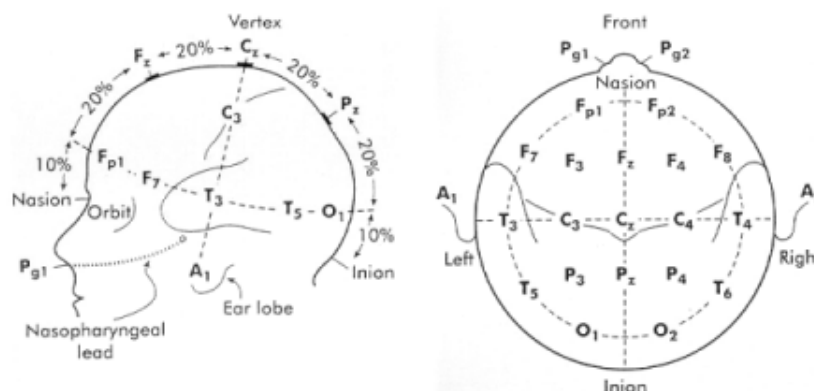


Figura 2.1: Sistema Internacional 10-20. Distribución de los electrodos sobre la superficie de la cabeza con las posiciones básicas obtenidas a partir del 10% y 20% de la medida total del cráneo[1].

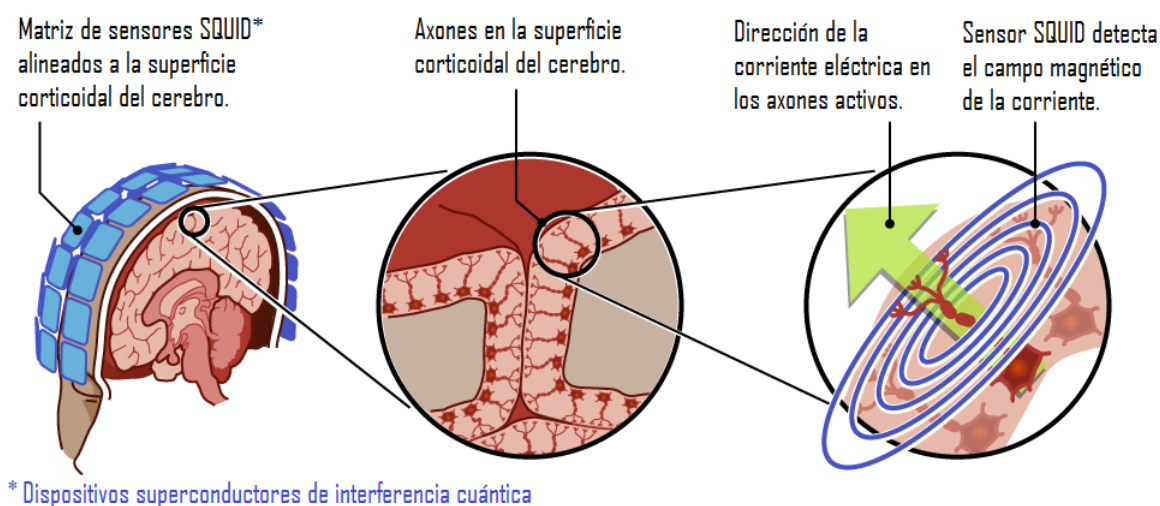


Figura 2.2: Detección de Campo Magnético mediante sensores SQUID.

Hasta aquí, el registro de PE no difiere del registro del EEG espontáneo. Sin embargo, cabe mencionar dos aspectos diferenciales en el proceso que nos permite obtener cada uno de ellos. La primera diferencia destacable entre los PE y el EEG espontáneo se refiere a su generación. Dado que un PE recoge actividad eléctrica cerebral provocada por la presentación de un estímulo externo, será necesaria bien la exposición de un estímulo o bien la realización de una tarea para que el PE aparezca. Dicho de otro modo: si no tenemos un suceso (o evento) no podremos provocar un PE. El segundo aspecto diferencial en los procedimientos para obtener los PE y el EEG tiene que ver con las diferencias que hay entre ellos en cuanto a su voltaje. A saber: en términos generales la amplitud está entre 1 y 20 microvoltios en los PE y entre 50 y 100 microvoltios para el EEG (Kutas y Van Petten, 1994). En consecuencia, resulta difícil, por no decir imposible, distinguir una fluctuación tan pequeña en voltaje a lo largo del EEG espontáneo. No obstante, existen distintos métodos para separarlos: el más utilizado consiste en el cálculo de un potencial promedio que permite detectar el PE deseado. Llegaremos a este potencial promedio registrando en primer lugar varios fragmentos (o épocas) del EEG continuo tras la presentación de un mismo estímulo y, posteriormente, promediándolos. La suposición que motiva esta técnica radica en lo siguiente: la actividad eléctrica que no está relacionada con el procesamiento del estímulo externo varía aleatoriamente a lo largo de las distintas épocas registradas, mientras que la actividad provocada por el estímulo se mantendrá constante. De esta manera, en el promedio, las fluctuaciones aleatorias en voltaje se igualarán a cero y se destacará una onda residual; esta actividad, en la que se han cancelado las fluctuaciones aleatorias, es el PE. Debido a las características propias del promediado, se debe trabajar con muchos ensayos para conseguir una adecuada razón señal/ruido, es decir, para obtener una mejor definición de la señal.

2.2.1. El problema del ruido.

El EEG nos dice que algo pasa, pero no el qué, y esto es a causa del ruido (ver Figura 2.3). Existe una forma de superar este problema y consiste en presentar repetidamente un determinado estímulo y, mediante software o hardware, limpiar (eliminar o filtrar) el

ruido que no procede de este estímulo.

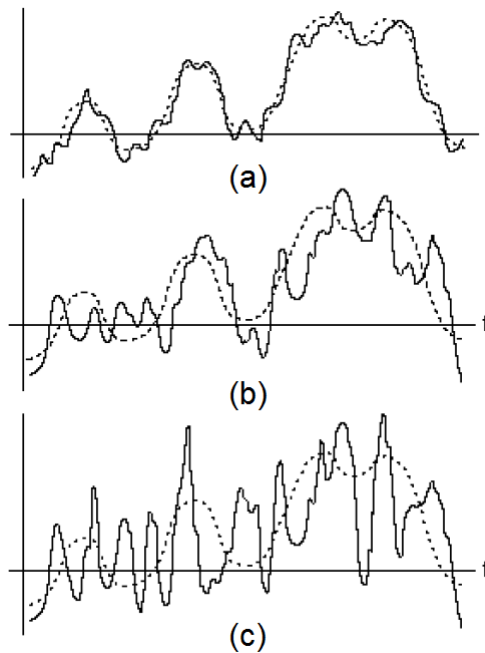


Figura 2.3: Señal contaminada por ruido. (a), (b) y (c), ruidos de pequeña, mediana y gran intensidad, respectivamente. Observe que cuanto más elevado es el nivel de ruido más distorsionada se encuentra la señal total [9].

Este filtro limpia la actividad extraña y mide el PE (que es lo que evoca el estímulo o señal). Usualmente, suele producirse gran cantidad de ruido en la representación de la señal debido a la sensibilidad de los sensores que se adhieren al cuero cabelludo, lo que hace indispensable el uso de este tipo de filtros para obtener una señal útil para su estudio.

Puesto que la relación señal/ruido del PE con respecto al EEG es pequeña, se hace necesario para su obtención el uso de técnicas como la mencionada anteriormente. Esta técnica supone que el PE y el EEG son estadísticamente independientes y que el PE está sincronizado con el estímulo. Las desventajas de la promediación es que se necesita un gran número de estimulaciones para obtener un sólo PE promedio, lo cual fatiga al sujeto y no permite el estudio de la variabilidad de los PE de estímulo a estímulo [10].

En un *Potencial Evocado Promedio* PEP se analizan formas de onda, amplitudes y latencia media. Se hace muy atractivo desde el punto de vista de investigación médica la obtención de la mejor estimación posible de un Potencial Evocado a partir de la respuesta a un número mínimo de repetición de los estímulos (como se observa en la Figura 2.4). En este caso se considera el empleo de un cancelador adaptivo de dos canales, con el cual se busca minimizar el error cuadrático medio de la señal del canal primario con respecto a la referencia procesada por un filtro adaptativo. El canal primario tiene PE + EEG. La referencia es una extensión periodica del PEP de pocos estímulos. Esta señal alimenta el filtro adaptativo FIR transversal, cuyos coeficientes se ajustan usando diferentes algoritmos:

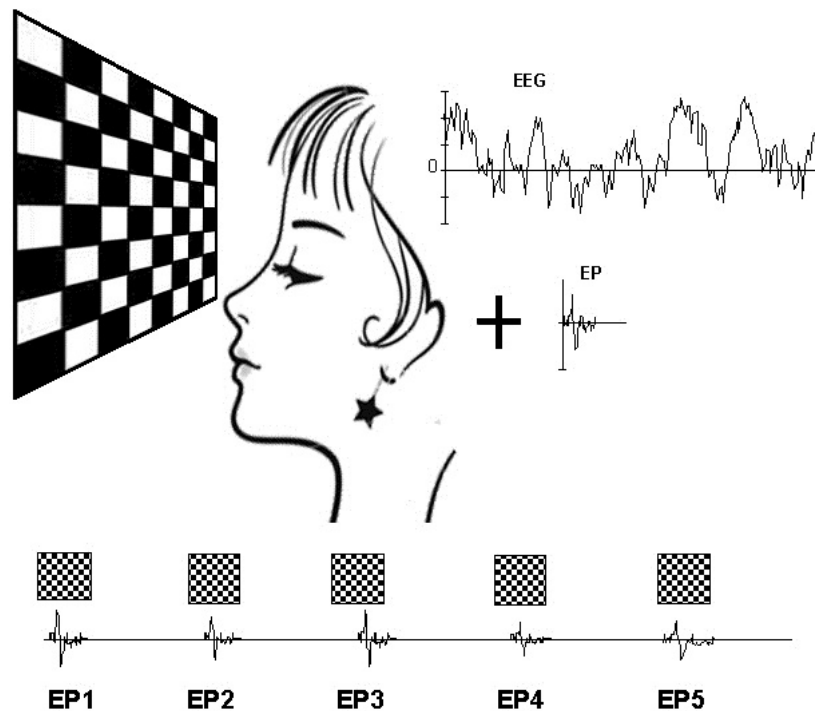


Figura 2.4: Ejemplo de promediación de ondas para la obtención del PEP.

Least Mean Squares, Carlos Dávila y Gram Schmidh [10].

Los experimentos se realizaron con señales simuladas obtenidas a partir de registros verdaderos obtenidos tras estimular visualmente a tres sujetos. La señales simuladas se construyen promediando las respuestas a 100 estímulos, lo que dio como resultado la señal patrón.

Sin variación de latencia, la señal primaria usada es una extensión periódica del patrón a la cual se le suma el EEG variando la potencia de éste. Para simular latencia variable, la señal primaria usada es una versión cuasi-periódica del patrón con cambios aleatorios de latencia (0-9 muestras). Debido a que el promedio de la señal primaria con latencia variable es una versión dispersa del patrón, la señal de referencia se construye de la siguiente forma:

1. Se calcula la función de correlación entre la señal primaria y el promedio original de 35 períodos.
2. De acuerdo al máximo de esta función se realizan dos mejoras:
 - a) Se alinea cada período de la señal primaria y se obtiene un PE promedio mejorado. Este se llamará *PE promedio correlacionado*.
 - b) Este *PE promedio correlacionado* se repite cuasiperiódicamente según marcadores en el tiempo correspondientes a los máximos de la función de correlación entre la señal primaria y el PE promedio. Para evaluar el comportamiento del sistema utilizamos el índice de correlación (IC) entre el patrón y el estimado, el cual se define:

$$IC = R_{xy}(MAX)/\sqrt{P_x P_y}$$

$$\text{donde } \begin{cases} R_{xy} = \text{Correlación cruzada} \\ P_x = \text{Potencia de la señal x} \end{cases} \quad (2.1)$$

Se realizaron diferentes pruebas variando la relación señal/ruido de la señal primaria. El valor del IC obtenido para el estimado N^o25 usando la técnica de cancelación puede compararse con el IC para el promedio convencional y el IC del promedio correlacionado. El resultado se muestra en la Figura 2.5. Se puede observar que la alineación previa a la promediación mejora el PE promedio; sin embargo, la técnica propuesta produce aún mejores resultados sin olvidar que además produce el PE debido a cada estímulo.

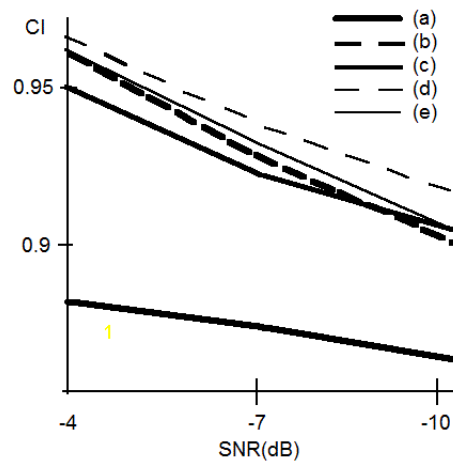


Figura 2.5: Índice de Correlación (CI) versus SNR para: a) Promedio, b) Promedio correlacionado, c) LMS, d) CDA, e) RMGS.

La búsqueda de mejores estimados de los PE individuales y con un menor número de estímulos ha motivado diferentes trabajos en este área. La técnica presentada produce, con un menor número de estímulos, mejores estimados de los PE individuales. Esto permitirá el análisis de las variaciones entre las respuestas a los estímulos secuencialmente en el tiempo, ayudando al terapeuta a realizar un mejor diagnóstico [10].

2.3 Características.

Un componente ha de definirse por una combinación de su polaridad, latencia, topografía y sensibilidad a las características de la manipulación experimental. Cabe destacar que las características primera y tercera recogen información sobre la fuente fisiológica, mientras que las otras dos se refieren a la función psicológica [1].

- En primer lugar, en lo que se refiere a la **polaridad**, los componentes pueden ser de dos tipos: positivos o negativos (ver Figura 2.6). En este sentido, denominaremos:

- P a los componentes con polaridad positiva,
- N a los que tengan polaridad negativa.

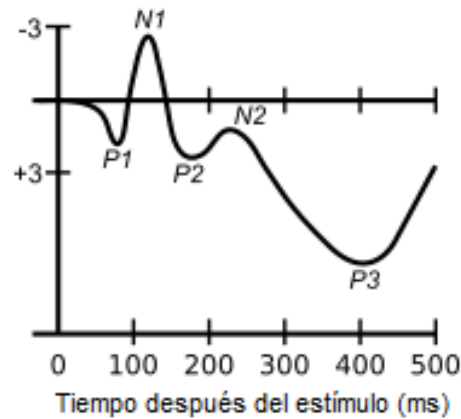


Figura 2.6: Polaridad de componentes. Positivos (P) y negativos (N).

- En segundo lugar, con respecto a la **latencia**, ésta suele medirse tomando el tiempo en milisegundos desde la presentación del estímulo hasta la aparición del pico (o del valle); esto es, hasta el punto de máxima o mínima amplitud dentro de una ventana de latencia concreta. Así, por ejemplo, el componente N400 es una onda negativa que presenta un pico aproximadamente a los 400 milisegundos después de la presentación del estímulo. Esta denominación que recoge la latencia del componente tiene un problema, ya que, en muchas ocasiones, se utiliza una terminología de este tipo para identificar un componente que puede presentar variaciones importantes en cuanto a su latencia. Tal es el caso del componente P300, un pico de polaridad positiva que ha sido ampliamente investigado (en la literatura encontramos P300 incluso en un rango de latencia entre 500 y 700 ms, dependiendo del tipo de estímulo, de los sujetos o de la tarea). Es por esta razón por la que algunos investigadores prefieren designar los componentes con otro término. Proponen mantener la polaridad pero con el añadido de un número que indique el orden de aparición del pico. Por ejemplo: P3 se referirá a un pico positivo que aparece en tercer lugar; N2 designa el segundo pico negativo después de la presentación del estímulo.
- La tercera característica definitoria de los componentes, según la propuesta de Donchin, Ritter y McCallum (1978), es su **topografía**, es decir, su distribución en el cuero cabelludo. Hemos comentado anteriormente que el registro de la actividad eléctrica se realiza en varias localizaciones, lo que nos permite, entre otras cosas, detectar si existe un lugar en concreto donde aparece un determinado componente y si hay diferencias hemisféricas. No obstante, debemos tener muy en cuenta lo siguiente: las fluctuaciones en el voltaje recogidas con un electrodo sobre el cuero cabelludo no han de tomarse como actividad originada por el tejido cerebral directamente subyacente a esa localización, es decir, que la topografía sobre el cuero cabelludo no suministra un mapa de la localización neuronal. De hecho, la actividad generada en

un área concreta del cerebro puede haberse registrado en una localización situada a una considerable distancia de su generador.

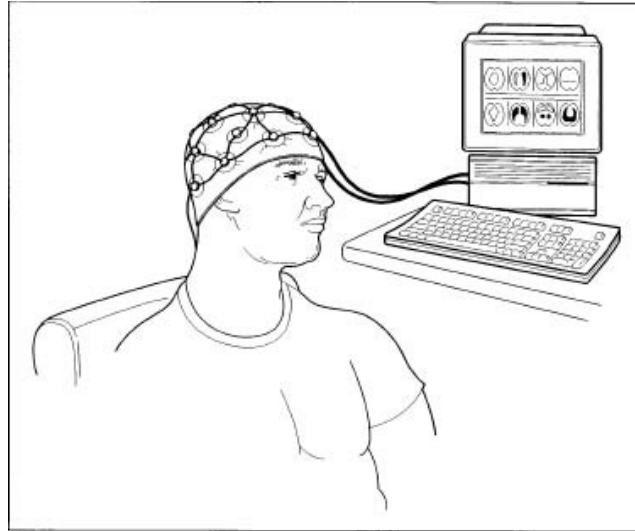


Figura 2.7: Distribución de los sensores por el cuero cabelludo.

- Finalmente, la última característica definitoria de un componente hace referencia a su **sensibilidad** respecto de una determinada manipulación experimental. Este es el criterio más difícil, puesto que existen componentes bastante inespecíficos. El caso más notorio es el de la familia de los P300, ondas que aparecen utilizando sucesos muy diversos. Otro componente controvertido es el N400, cuya especificidad como indicador del procesamiento semántico también se ha puesto en duda. A pesar de esto, la sensibilidad a determinadas manipulaciones experimentales sigue siendo un elemento fundamental para separar componentes.

Recientemente, se ha puesto de manifiesto el problema que supone la ausencia de consenso por parte de los investigadores para designar los distintos componentes. Estos autores distinguen entre lo que denominan terminología observacional y terminología teórica: la primera consiste en designar un componente en función de las características físicas de la onda medida en un grupo de datos, mientras que la segunda se fundamenta en el tipo de proceso psicofisiológico que supuestamente la origina. Afirman, además, que ante algunos estudios de PE cognitivos resulta difícil discernir si los investigadores están utilizando una terminología observacional o una teórica para referirse a un determinado componente. Sugieren, por tanto, que sería conveniente separar ambas nomenclaturas y su propuesta consiste en colocar una línea sobre el nombre del componente cuando nos estamos refiriendo al mismo en sentido teórico, por ejemplo, $\overline{P300}$. La revisión teórica que hemos llevado a cabo sobre distintos componentes de los potenciales evocados nos ha confirmado la idoneidad de la indicación de estos autores, puesto que en muchas ocasiones al leer los artículos no queda claro a qué se refieren exactamente los investigadores cuando designan un componente. [1]

2.4 Clasificación.

Los potenciales evocados se pueden clasificar según una gran variedad de criterios. En nuestro caso lo haremos en referencia al tipo de estímulo. Cabe destacar que, aunque se puede hacer una clasificación englobando tres grandes grupos (visuales, auditivos y somatosensoriales), para la realización de este proyecto solo se estudiarán los dos primeros.

2.4.1. Potenciales Evocados Visuales.

Los potenciales evocados visuales (PEV) resultan de los cambios producidos en la actividad bioeléctrica cerebral tras estimulación luminosa. El estímulo más frecuentemente utilizado para obtener PEV consiste en una imagen con forma de tablero de ajedrez (ver Figura 2.8), compuesta por una serie de cuadros blancos y negros que van alternándose (PEV-pattern) de forma periódica. Consigue evocar potenciales grandes y reproducibles aunque precisa la colaboración del paciente [11].



Figura 2.8: Test de Potencial Evocado Visual en bebés.

En pacientes no colaboradores o que no consiguen ver la pantalla con el tablero, se utilizan otros estímulos como destellos luminosos. Estos producen respuestas evocadas con gran variabilidad inter-individual, en morfología y latencias, por lo que únicamente sirven para determinar si llega el estímulo luminoso a la corteza cerebral y para comparar la respuesta de ambos ojos en busca de asimetrías.

Es la única prueba clínicamente objetiva para valorar el estado funcional del sistema visual. Registra las variaciones de potencial en la corteza occipital provocada por un estímulo sobre la retina. Por esta razón puede evaluar la función retinocortical en niños y pacientes neurológicos. También puede distinguir entre pacientes con ceguera psicológica y los que la padecen por una causa orgánica.

Los potenciales evocados visuales representan una exploración neurofisiológica muy sensible, ya que se alteran en una elevada proporción de pacientes con anomalías visuales, incluso en pacientes con afección subclínica de la vía visual, como ocurre en la esclerosis múltiple. Sin embargo, es una exploración poco específica a la hora de determinar el tipo de patología, ya que cualquier problema que se interponga entre el estímulo y el

registro en corteza occipital, puede provocar anomalías en los potenciales visuales siempre que cause suficiente disfunción visual (defecto de corrección óptica, catarata densa, retinopatía, glaucoma, neuropatía óptica, infarto cerebral, etc). Deben, por tanto, evaluarse con precaución y dentro de un contexto clínico.

Otra característica de enorme valor de los potenciales evocados visuales es que aportan datos cuantificables de latencia y amplitud. Esto nos permite identificar una disfunción en la vía visual, orientando si predominan los fenómenos desmielinizantes, con retraso de los potenciales (aumento de latencia) o si predomina un defecto de activación axonal en la vía visual (reducción de amplitud). Por otro lado, nos permite realizar un seguimiento evolutivo, pudiendo evaluar la posible eficacia de un tratamiento, o la progresión de una enfermedad.

2.4.2. Potenciales Evocados Auditivos.

Los potenciales evocados auditivos (PEA) son el resultado de registrar la actividad electroencefalográfica desencadenada a partir de la presentación de un estímulo acústico continuo y periódico en el tiempo (ver Figura 2.9). El estímulo es altamente específico en frecuencia, lo cual permite obtener respuestas selectivas. [12]



Figura 2.9: Test de Potencial Evocado Auditivo en bebés.

Una de las grandes ventajas de esta técnica es la forma en la que se presentan los estímulos, ya que nos permite valorar distintas frecuencias del audiograma de forma simultánea y en ambos oídos a la vez.

El potencial evocado auditivo (PEA) se provoca con la aplicación de un estímulo sonoro breve, conocido como “click”, por medio de la aplicación de frecuencias puras, proceso conocido como estimulación tonal, o mediante la estimulación con fonemas. Se originan por la actividad eléctrica que los estímulos provocan a lo largo de toda la trayectoria sensorial que incluye el oído, el nervio auditivo y las regiones de la corteza cerebral

relacionadas con la audición.

Las características del PEA dependen de factores asociados al estímulo, a la adquisición misma del PEA y al sujeto. Entre los primeros se cuentan la intensidad, duración, tipo de estímulo aplicado, la frecuencia de estimulación, tipo y ubicación del transductor, polaridad acústica y si el estímulo es estéreo o monofónico. Los asociados a la adquisición son de carácter técnico como el tipo de electrodos, ubicación específica de éstos y su proximidad a la fuente generadora, la amplificación de las señales eléctricas y los procesos a los que son sometidas como el filtrado, la promediación y procesamiento digital. Los factores vinculados al sujeto pueden ser de naturaleza patológica (pérdida de audición en el oído medio o la cóclea, disfunciones del octavo nervio, del tallo cerebral o de la corteza auditiva) y no patológica (edad, género, estado de alerta, temperatura corporal, actividad muscular y presencia de fármacos en el organismo, entre otros).

2.4.3. Potenciales Evocados Somatosensoriales.

La prueba de los potenciales evocados somatosensoriales (PES) es un procedimiento neurofisiológico sensible con el cual podemos, objetivamente, y sin riesgo para el paciente, evaluar la integridad de la vía somatosensorial desde los receptores cutáneos o nervios periféricos mielinizados (tipo IA) hasta la corteza somatosensorial primaria (ver Figura 2.10). Esto se determina midiendo las latencias absolutas y relativas de las ondas generadas a nivel del nervio periférico [13].



Figura 2.10: Test de Potencial Evocado Somatosensorial en adultos.

El objetivo consiste en detectar anomalías en la conducción midiendo la latencia absoluta y relativa de las ondas a nivel próximo al nervio periférico estimulado, entrada a la médula espinal y cortical en pacientes en los cuales el diagnóstico, pronóstico, curso y eficacia del tratamiento de trastornos que afecten primaria o secundariamente a las vías

somatosensoriales, pero que no se pueden diagnosticar solamente con el examen neurológico y otras pruebas neurodiagnósticas convencionales.

2.5 El componente P300.

El grupo de los componentes P300 se ha investigado extensamente, tanto desde un punto de vista teórico como desde una perspectiva clínica. Se trata de una serie de picos positivos los cuales, a pesar de presentar múltiples diferencias entre ellos, también presentan distintos elementos en común [1].

- En primer lugar, suelen aparecer ante estímulos inesperados que son relevantes para la tarea que ha de realizar el sujeto.
- En segundo, su latencia varía dependiendo de la dificultad de la tarea.
- En tercero, su amplitud se ve modulada por el grado de expectación que el sujeto tiene de un determinado estímulo.

El componente P300 se ha utilizado en una gran variedad de aplicaciones teóricas, empíricas y clínicas. En particular, se considera como un índice objetivo relativo a la capacidad de la memoria de trabajo, de forma que su amplitud informa de la asignación de los recursos atencionales y su latencia es un indicador del tiempo de evaluación del estímulo.

Se piensa que la misma onda P300 está compuesta por dos ondas secundarias conocidas como “señales P3a y P3b”. Estos componentes responden individualmente a diferentes estímulos y se ha sugerido que la onda P3a se origina en los mecanismos de atención frontal dirigidos por estímulo durante el procesamiento de tareas, mientras que la P3b se origina en la actividad parietal-temporal asociada con la atención y parece estar relacionada con el consiguiente procesamiento en la memoria. Algunas veces se denomina potencial evocado “diana (target)” a la P3b, mientras que la P3a sería el potencial “no diana (non-target)” [14].

En la Figura 2.11 se comparan los patrones de actividad cerebral en respuesta a los ensayos precedidos por un tono que indica la necesidad de cambiar de criterio atencional (línea continua), frente a los ensayos en los que el sujeto debe clasificar por el mismo criterio que en ensayos previos (línea discontinua) [15].

2.5.1. Obtención de la respuesta.

La señal P300 es un registro agregado del procesamiento definido por una gran cantidad de neuronas. Aunque es una técnica típicamente no invasiva, se pueden tomar muestras de partes de la señal más directamente de ciertas partes del cerebro mediante electrodos. Esta metodología permite aislar y registrar localmente un área sin el ruido de otras señales adquiridas a través de electrodos sobre la piel del cráneo. En la práctica, la forma de onda

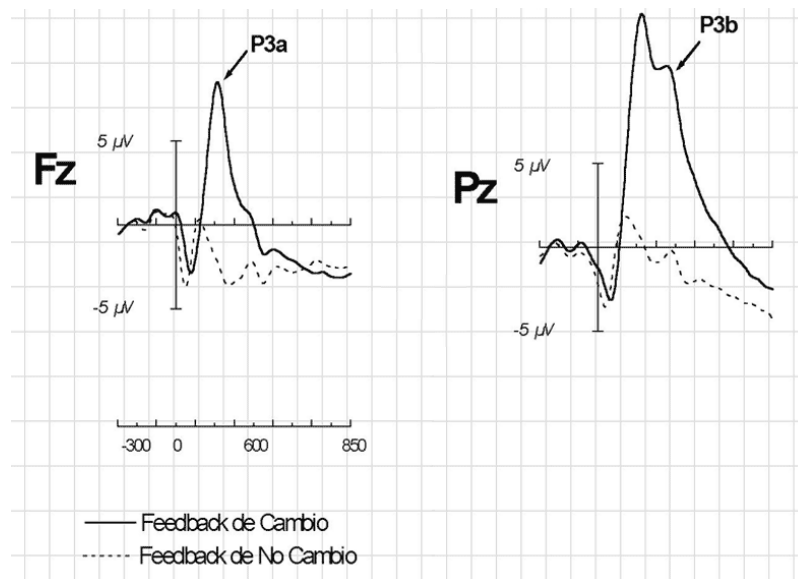


Figura 2.11: Representación en los ejes cartesianos de los componentes P3a con un máximo en el electrodo Fz (figura superior) y P3b con un máximo en el electrodo Pz (figura inferior) [15].

de la P300 se debe evocar utilizando un estímulo provisto por alguna de las modalidades sensoriales [14]. Un procedimiento típico es el *paradigma odd-ball*, donde dos estímulos, uno de mayor probabilidad que el otro, se presentan en orden aleatorio. El sujeto debe discriminar el estímulo infrecuente (diana), del frecuente (estándar), bien apretando un botón de respuesta o bien contando su ocurrencia mentalmente [1].

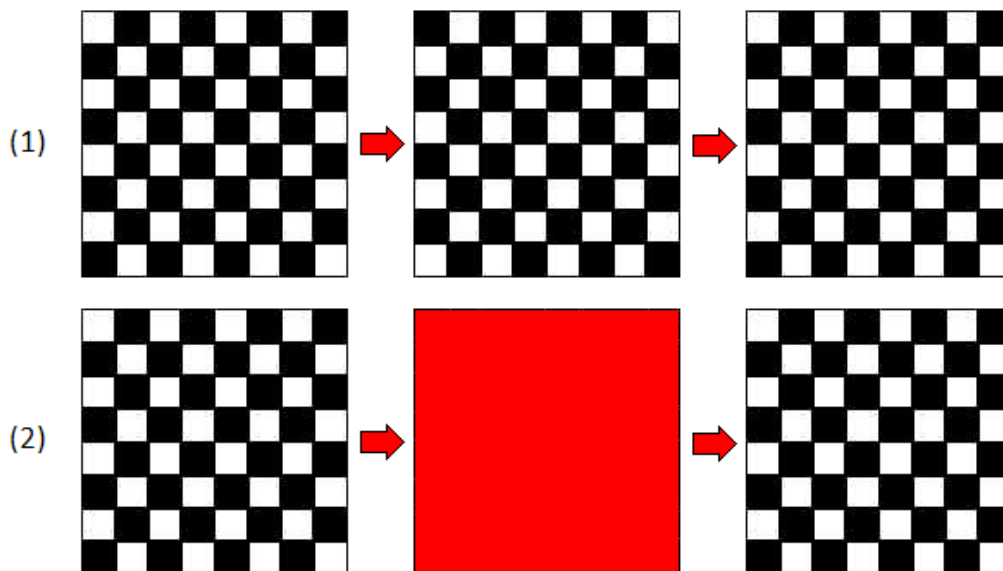


Figura 2.12: Paradigma Odd-ball. Estímulos frecuente (1) e infrecuente (2).

El *paradigma odd-ball* clásico ha sufrido muchas variaciones, pero al final la mayor parte de los protocolos utilizados para evocar un componente P300 implica alguna forma

de hacer consciente una situación o una toma de decisiones. Se necesita ejercitar el proceso de atención en estos protocolos [14].

2.5.2. Parámetros del P300.

En la valoración del componente P300 se tienen en cuenta tres parámetros [16]:

- La **topografía** o distribución de los potenciales en las distintas áreas cerebrales.
- La **amplitud** o diferencia en mV entre el punto de máxima altura de la onda y la media de los valores obtenidos durante la línea de base preestímulo.
- La **latencia** o tiempo en milisegundos desde el inicio del estímulo hasta la producción de un cambio eléctrico o respuesta. El punto de la respuesta que suele medirse para establecer la latencia es su inicio, aunque en ocasiones se utiliza el pico. En general, el pico se utiliza como criterio cuando el inicio es difícil de determinar o aparece enmascarado por una respuesta anterior.

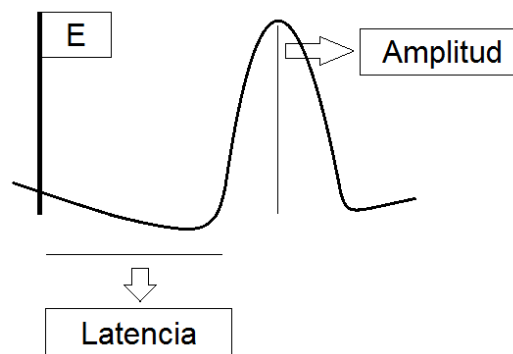


Figura 2.13: Parámetros del P300. Amplitud y Latencia

Se ha demostrado que en personas ancianas, la onda P300 tiene una latencia mayor y una amplitud menor, además de una diferente distribución en el cuero cabelludo. Por otra parte, en niños de edades comprendidas en los 6 y 15 años existe una disminución de la latencia de la onda P300 a medida que el niño tiene más edad, llegando algunos autores a relacionar el desarrollo de la memoria de los niños con la disminución de la latencia de la P300, lo que sugiere que la maduración de la memoria inmediata está ligada al aumento de la velocidad de identificación. Los generadores de la onda P300 no han sido determinados con precisión. La aproximación neurofisiológica busca precisar las estructuras cerebrales que sostienen los componentes funcionales y cuya activación se traduce sobre el cuero cabelludo por una sucesión de deflexiones positivas y negativas.

2.5.3. Aplicaciones.

Puesto que el deterioro cognitivo a veces está correlacionado con modificaciones en la P300, la forma de onda a veces se puede utilizar como medida para la eficacia de

distintos tratamientos para la rehabilitación neurocognitiva. Se ha sugerido su uso como marcador clínico precisamente por estas razones. Hay un amplio rango de usos para la P300 en la investigación científica, que va desde el estudio de la depresión y la drogadicción a los trastornos de ansiedad (trastorno obsesivo-compulsivo, trastorno por estrés post-traumático, etc.) y también se ha propuesto su uso en interfaces computadora-cerebro [14].

Sin embargo, para este proyecto nos centraremos en el uso de este componente para el estudio del TDAH ya que está relacionado con una gran variedad de actividades cognitivas, como atención selectiva, respuesta de orientación, probabilidad del estímulo y la toma de decisiones. El elemento común a estas actividades es el procesamiento activo por parte del sujeto de la información proporcionada por el estímulo. La amplitud de la P300 no se ve afectada por las propiedades físicas, como la modalidad o intensidad, pero sí está influida por variables psicológicamente importantes, como la frecuencia de aparición del estímulo diana, la dificultad de la tarea, el intervalo interestímulo y el reparto de los recursos de atención. Por otra parte la latencia de la P300 se considera una medida del tiempo de evaluación del estímulo recibido, indicando la cronometría de los acontecimientos mentales antes de que la respuesta sea seleccionada o producida. Cuanto más prolongada es la latencia mayor es el tiempo empleado en el procesamiento de la información. Esta medida puede utilizarse como un índice de la secuencia del proceso de evaluación de los estímulos independiente del tiempo necesario para la ejecución de una respuesta motora adecuada [16].

Algunas investigaciones se han realizado comparando la actividad electroencefalográfica de niños control con niños con TDAH mientras realizan una tarea [14]. Como objetivos principales, la mayoría de estos trabajos se han centrado en:

- Medir principalmente la amplitud y la latencia de la P300, la cual es generada cuando los sujetos atienden y discriminan eventos o se produce un proceso de evaluación estimular.
- Examinar qué sucede con los procesamientos previos a la misma, (N100 y la negatividad del procesamiento, N200).
- Comprobar si la ejecución en la tarea de los niños con TDAH es peor que la de los niños control.
- Seguimiento evolutivo de pacientes tras la administración de un protocolo de rehabilitación. Esta línea de tratamiento del déficit atencional es la que más se ha desarrollado con el avenimiento tecnológico y de la neurociencia, en especial de la neuropsicología y es donde han surgido los tratamientos más eficaces, sistemáticos e innovadores basados en la estimulación de las distintas funciones cognitivas del cerebro humano. La realización de diversos ejercicios permite la estimulación de áreas cerebrales implicadas en este trastorno, permitiendo mejorar la alerta, la concentración (atención sostenida), planificación, organización, verificación de resultados, cálculos, capacidad de razonamiento, selección de estímulos, etc., logrando mejoras significativas en la atención o en la realización de una determinada tarea, además de mejoras en memoria visual y auditiva y en memoria operativa en general.

3

Especificación de *AWARD Potentials*

En este capítulo se detallará el plan de trabajo (indicando el inicio y fin de cada una de las tareas realizadas) y se especificarán los diagramas de casos de uso así como el diagrama de clases informal resultado de la fase de análisis.

3.1 Plan de trabajo.

El objetivo de esta sección consiste en mostrar el tiempo de dedicación previsto para las diferentes tareas necesarias para implementar y documentar nuestra aplicación a lo largo de un tiempo total determinado, ya que nos propusimos el mes de Julio como fecha límite para la finalización de este proyecto.

El método elegido para la planificación temporal de nuestro proyecto ha sido el diagrama de Gantt. Este tipo de gráfico permite identificar la actividad en la que se estará utilizando cada uno de los recursos así como la duración, de tal modo que puedan evitarse periodos ociosos innecesarios y se de también al administrador una visión completa de la utilización de los recursos que se encuentran bajo su supervisión.

En el siguiente diagrama de Gantt se muestra el origen y el final de las diferentes unidades mínimas de trabajo y los grupos de tareas que componen nuestro plan de trabajo:

- En el eje Horizontal, un calendario definido en términos de la unidad más adecuada al trabajo que se va a ejecutar, en nuestro caso semanas, que van desde la última semana de Noviembre de 2010 hasta la tercera semana de Julio de 2011.
- En el eje Vertical, las actividades que constituyen el trabajo a ejecutar. A cada actividad se hace corresponder una línea horizontal cuya longitud es proporcional a su duración.

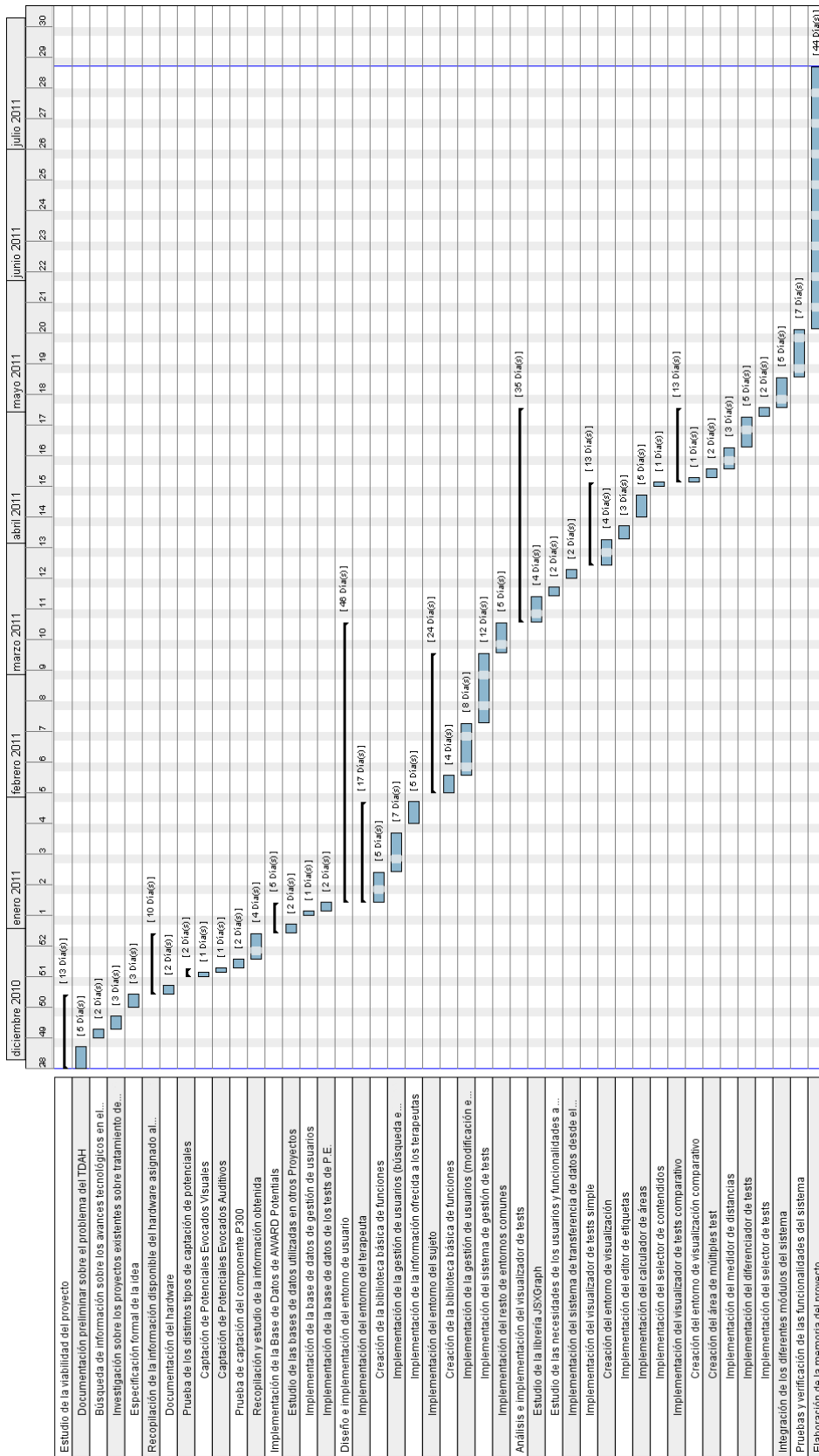


Figura 3.1: Diagrama de Gantt (planificación temporal).

3.2 Funcionalidad del sistema y Casos de uso.

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. En la realización de nuestro diagrama de casos de uso hemos usado tres elementos básicos:

- Actores: Los actores representan un tipo de usuario del sistema. Un actor en un diagrama de casos de uso representa un rol que alguien puede estar adoptando, no un individuo particular.
- Caso de uso: Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando.
- Asociaciones: Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo tal caso de uso.

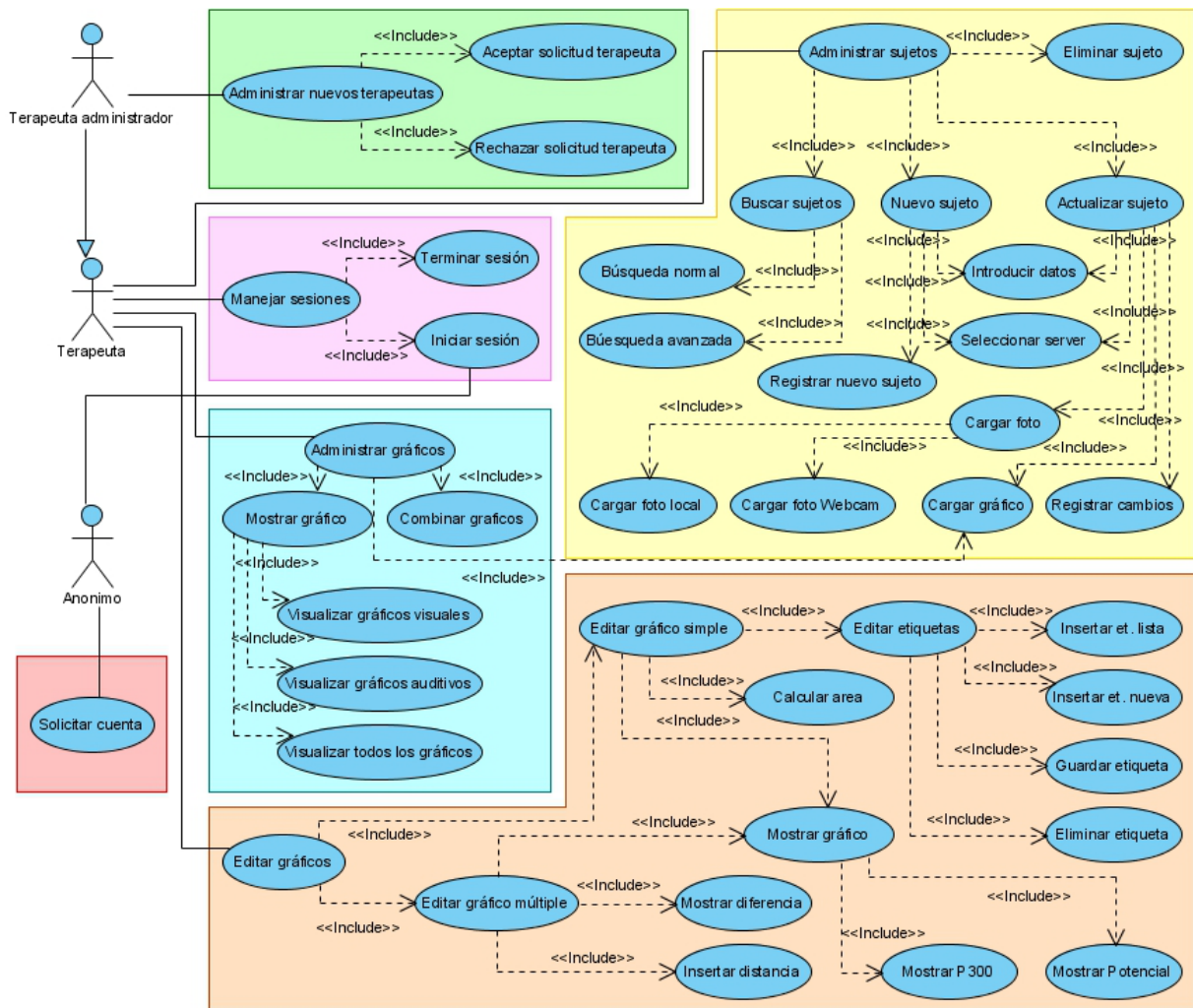


Figura 3.2: Diagrama de Casos de Uso completo.

3.2.1. Actores del diagrama de casos de uso.

En nuestro sistema existen tres tipos de actores claramente diferenciados:

- **Anónimo.** Este actor es el que visita la aplicación sin identificarse. Toda persona que inicia por primera vez la aplicación es un actor “Anónimo”. Este tipo de actor sólo tiene dos casos de uso, iniciar sesión (si ya está registrado) para pasar a ser un actor de otro tipo, o bien, solicitar cuenta para obtener los datos de inicio de sesión.
- **Terapeuta.** Este actor ya está registrado en la aplicación y además ha accedido con su login y password a la misma, por lo que tiene permisos para realizar todas las acciones que se le permiten a los actores de su clase.
- **Terapeuta administrador.** Este tipo de actor hereda los casos de uso del actor “Terapeuta” y además tiene algunos propios. Se trata de los casos de uso que permiten administrar las solicitudes de posibles nuevos terapeutas (actores Anónimos).

3.2.2. Casos de uso.

A continuación se describe cada uno de los casos de uso que se muestran en el diagrama general. Están divididos en las principales asociaciones para una mayor comodidad de visualización.

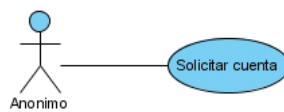


Figura 3.3: Diagrama de Casos de Uso “Solicitar cuenta”.

- **Solicitar cuenta** (Figura 3.3). El actor “Anónimo” puede solicitar una cuenta de tipo Terapeuta en la aplicación mediante esta acción. Una vez solicitada la cuenta de usuario deberá esperar a que un actor “Terapeuta administrador” acepte o rechace su petición.

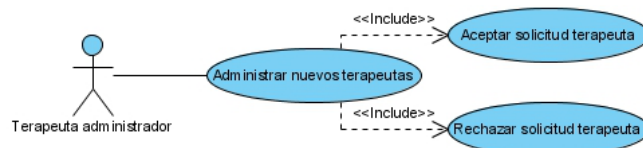


Figura 3.4: Diagrama de Casos de Uso “Administrar nuevos terapeutas”.

- **Administrar nuevos terapeutas** (Figura 3.4). Este caso de uso engloba las acciones que puede realizar un Terapeuta administrador con las solicitudes entrantes de los actores Anónimos.

- *Aceptar solicitud terapeuta.* El Terapeuta administrador puede aceptar la solicitud de alta en el sistema del actor Anónimo, con lo cual, se introducirían los datos pertinentes de este actor en la base de datos del sistema con lo que ,consecuentemente, podría disfrutar de las ventajas en cuanto a funcionalidades de las que disponen los Terapeutas en la aplicación.
- *Rechazar solicitud terapeuta.* El Terapeuta administrador puede rechazar la solicitud enviada por el actor Anónimo, con lo cual seguirá sin poder acceder a la aplicación mediante el rol de Terapeuta.

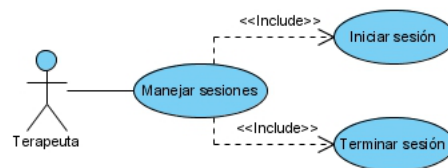


Figura 3.5: Diagrama de Casos de Uso “Manejar sesiones”.

- **Manejar sesiones** (Figura 3.5). Aquí se engloban las acciones para entrar y salir del sistema.
 - *Iniciar sesión.* Un actor Anónimo introduce un nombre de usuario y contraseña y, si éstos se encuentran en la base de datos y son válidos, este actor entra en la aplicación como “Terapeuta”.
 - *Terminar sesión.* Un actor Terapeuta o Terapeuta administrador puede terminar su sesión y pasar a ser un actor Anónimo. Esta acción es útil si se quiere cambiar de usuario o, simplemente, si ya no se desea seguir trabajando con la aplicación.
- **Administrar sujetos** (Figura 3.6). Un Terapeuta puede realizar diferentes acciones con los sujetos que se encuentran asociados a él en la base de datos.
 - *Buscar sujetos.* Se pueden realizar dos tipos de búsqueda.
 - Búsqueda normal. Este tipo de búsqueda es el más habitual. El Terapeuta deberá introducir el nombre y/o apellidos del sujeto con el que desea trabajar. Se mostrarán todos los sujetos coincidentes total o parcialmente con los datos introducidos para los campos mencionados.
 - Búsqueda avanzada. Con esta búsqueda se pueden realizar consultas en la base de datos mediante cualquier campo perteneciente al sujeto que estemos buscando. Se podrán realizar búsquedas tanto por un único campo, como por todos los campos disponibles a tal fin.
 - *Nuevo sujeto.* Este caso de uso se encarga de registrar nuevos sujetos asociados a un terapeuta en la base de datos.

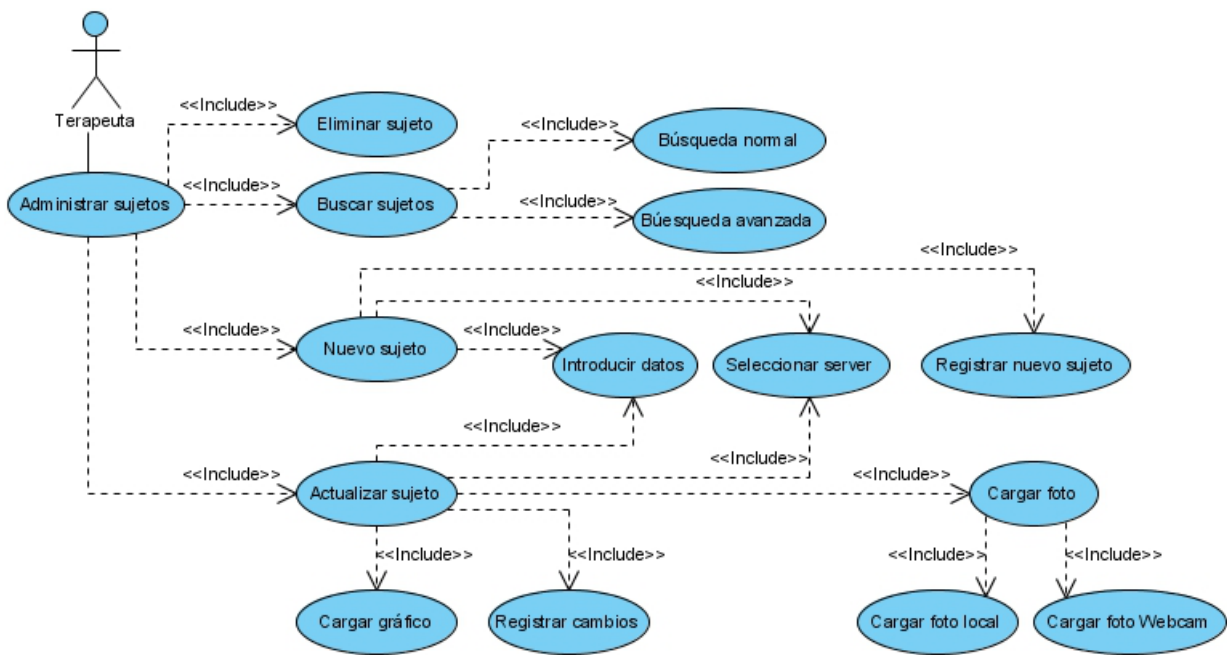


Figura 3.6: Diagrama de Casos de Uso “Administrar sujetos”.

- Introducir datos. El Terapeuta podrá introducir los campos que desee sobre la información personal de un sujeto. Hay un campo obligatorio denominado “código” que funciona como la clave primaria de ese sujeto en la base de datos. La aplicación avisará al terapeuta en ese mismo instante de si el código está repetido o es válido para la inclusión del nuevo sujeto en la base de datos.
- Seleccionar server. Se podrán cargar los datos del sujeto mediante la importación de los mismos desde servidores pertenecientes a otros proyectos *AWARD*.
- Registrar nuevo sujeto. Una vez introducidos los datos pertinentes, se podrán introducir los nuevos datos del sujeto en la base de datos.
- *Actualizar sujeto*. Los datos de los sujetos asociados al terapeuta podrán ser modificados mediante las siguientes acciones.
 - Cargar foto. Se puede introducir en el sistema una fotografía del sujeto. Esta acción se puede llevar a cabo de las siguientes dos formas:
 - ◊ Cargar foto local. El terapeuta podrá seleccionar un archivo de imagen con la fotografía del sujeto y además recortarla desde la misma aplicación si es preciso para así poder almacenarla en el sistema y en la base de datos.
 - ◊ Cargar foto Webcam. También se podrá capturar la imagen en tiempo real si se dispone de una webcam. El terapeuta podrá realizar esta acción sin necesidad de salir del sistema para luego cargar la fotografía como archivo local (caso de uso anterior) ya que el sistema implementa todo lo necesario para el uso de este hardware así como el almacenamiento posterior de la captura.

- Cargar gráfico. El terapeuta podrá cargar un archivo de gráfico exportado (potencial evocado) mediante el software *MaxG EP-1* y éste será introducido automáticamente como un nuevo registro en la base de datos asociado al terapeuta que realizó la subida. El terapeuta deberá indicar si el gráfico introducido en el sistema corresponde a un Potencial Evocado de tipo Visual o Auditivo. Tras esto, el gráfico está disponible para las acciones de edición que explicaremos más adelante.
- Registrar cambios. El terapeuta podrá guardar todos los cambios realizados en el perfil de cualquiera de sus sujetos.
- *Eliminar sujeto*. Los sujetos podrán ser eliminados del sistema y, consecuentemente, de la base de datos. Del mismo modo, al eliminar un sujeto, también se eliminarán todos los gráficos asociados al mismo.

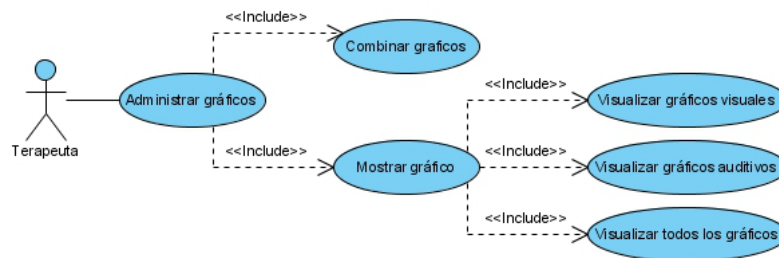


Figura 3.7: Diagrama de Casos de Uso “Administrar gráficos”.

- **Administrar gráficos** (Figura 3.7). Un terapeuta, tras seleccionar un sujeto determinado asociado a él y existente en la base de datos, puede visualizar y administrar los gráficos pertenecientes a este dicho sujeto.
 - *Mostrar gráficos*. El terapeuta tendrá la opción de visualizar los gráficos de un determinado sujeto desde tres perspectivas distintas.
 - Visualizar gráficos visuales. Se mostrarán todos los gráficos de potenciales visuales pertenecientes al sujeto seleccionado.
 - Visualizar gráficos auditivos. Se mostrarán todos los gráficos de potenciales auditivos pertenecientes al sujeto seleccionado.
 - Visualizar todos los gráficos. Se mostrarán todos los gráficos pertenecientes al sujeto seleccionado.
 - *Combinar gráficos*. Una vez visualizados los gráficos pertenecientes a un sujeto, el terapeuta podrá combinar cuantos gráficos desee, para así poder editarlos en su conjunto (edición de gráficos múltiples).
- **Editar gráficos** (Figura 3.8). Las funciones principales del sistema son las encargadas de la edición de gráficos de los tests de PE (potenciales evocados). Se podrán editar gráficos de forma individual (Editar gráfico simple) o varios al mismo tiempo (editar gráfico múltiple) si el terapeuta necesita realizar algún tipo de comparativa entre distintos tests realizados en distintos instantes de tiempo.

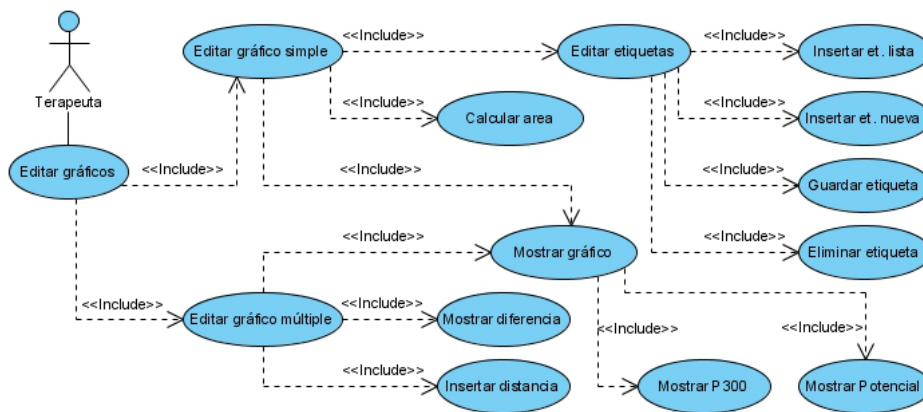


Figura 3.8: Diagrama de Casos de Uso “Editar gráficos”.

- *Editar gráfico simple.* El terapeuta edita un sólo gráfico con la intención de obtener resultados sin realizar comparativas evolutivas o del componente P300. Esta forma de edición tiene distintas funcionalidades o casos de uso que se explican a continuación.
 - Editar etiquetas. El terapeuta podrá insertar etiquetas en el gráfico y posicionarlas en las coordenadas que desee. Estas etiquetas podrán ser nombradas según diferentes criterios y guardadas o eliminadas del sistema.
 - ◇ Insertar etiqueta de la lista. Se inserta una etiqueta cuyo nombre pertenece a una lista de un menú desplegable. Estos nombres de etiquetas son los más comunes para asignar a componentes de potenciales evocados (P100, P300, ...).
 - ◇ Insertar etiqueta nueva. El terapeuta podrá insertar una etiqueta cuyo nombre es elegido por él mismo.
 - ◇ Guardar etiqueta. La etiqueta seleccionada es guardada en la base de datos y estará asociada a la gráfica con la que se está trabajando. Una vez cargado la gráfica tras esta operación, las etiquetas guardadas con anterioridad se mostrarán inmediatamente.
 - ◇ Eliminar etiqueta. El terapeuta podrá borrar una etiqueta seleccionada del gráfico. Esta acción, además de borrar la etiqueta del entorno visual, también la borrará de la base de datos.
 - Calcular área. El terapeuta podrá calcular el área formada por el gráfico entre dos puntos seleccionados del eje X a su elección. Estos puntos podrán ser modificados por el terapeuta de forma interactiva haciendo que el área se recalcule automáticamente tras los cambios efectuados.
 - Mostrar gráfico. El editor de gráfico simple permite al terapeuta distintas formas de visualización relacionadas con el gráfico que se muestra en ese momento en el editor. El terapeuta podrá elegir entre visualizar el potencial aislado, mostrar el P300 asociado a ese potencial de forma aislada, o bien, la combinación de ambos.
 - ◇ Mostrar Potencial. El terapeuta podrá visualizar el gráfico del potencial de forma aislada.

- ◊ Mostrar P300. El terapeuta podrá visualizar el gráfico del P300 asociado al potencial de forma aislada.
- ◊ Mostrar Ambos. El terapeuta podrá combinar el potencial con su P300 para así pasar a usar el editor de gráfico múltiple cuyo caso de uso se explica a continuación.
- *Editar gráfico múltiple.* Si el terapeuta desea realizar una comparativa evolutiva de los test de un paciente o bien, simplemente desea ver un potencial junto a su P300 y poder realizar distintas acciones con ellos, puede hacerlo gracias al editor de gráfico múltiple, cuyas principales funcionalidades pasamos a explicar:
 - Mostrar diferencia. El terapeuta podrá visualizar la diferencia de dos o más gráficos seleccionados previamente. Esta diferencia es el resultado de restar punto a punto el valor del eje Y de cada test. Esta información es realmente útil para el terapeuta ya que permite obtener la diferencia real del componente P300 de un potencial del que se han obtenido los gráficos frecuentes e infrecuentes.
 - Insertar distancia. Esta función permite al terapeuta calcular la distancia entre dos puntos o componentes de dos test distintos (o del mismo).

3.3 Diagrama de clases informal.

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para realizar su tarea en vez de objetos del sistema o propios de un modelo de diseño.

El diagrama de clases puede utilizarse con distintos fines en distintas etapas del proceso de desarrollo. Durante la etapa de análisis, el “modelo de dominio” es encargado de mostrar el conjunto de clases conceptuales del problema y las relaciones presentes entre sí. Por otro lado, durante la etapa de diseño, el modelo de diseño determina los futuros componentes de software (clases) y sus relaciones entre ellos.

En la figura 3.9 se puede ver el diagrama de clases informal (perteneciente a la fase de análisis) de nuestro sistema, compuesto por cuatro clases principales (Terapeuta, Sujetos, Gráficos y Tareas).

Como se puede observar, en el diagrama se muestran las relaciones entre las entidades principales, es decir, Terapeuta, Sujeto, Gráfico y Tarea. De forma resumida, el sistema va a permitir almacenar datos de los Sujetos a los que ha atendido de forma particular un

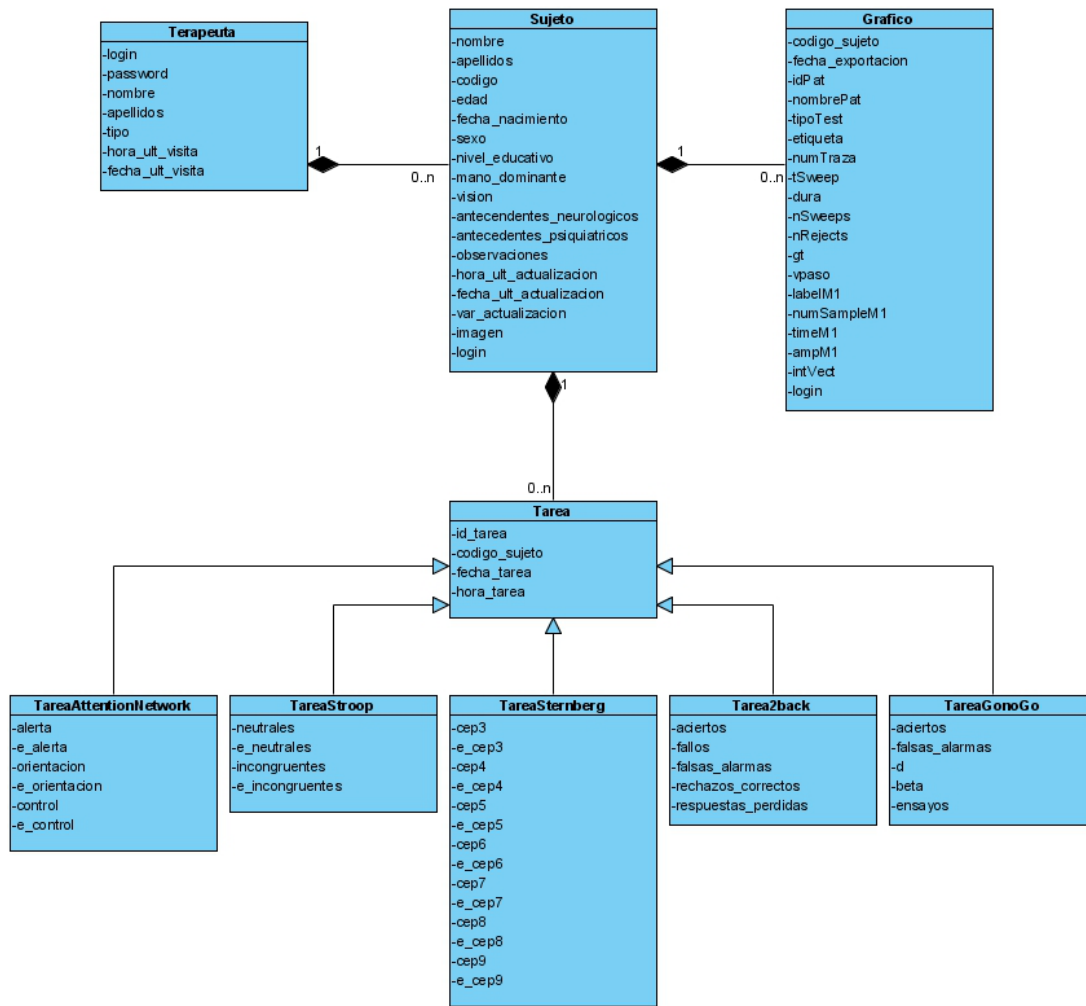


Figura 3.9: Diagrama de Clases (Fase de análisis).

determinado Terapeuta, que puede ser un profesional de la Psicología o la Neurofisiología. Para cada sujeto, además de almacenar sus datos personales así otros datos de interés, se permite el registro del resultado obtenido en distintas sesiones de captura de Potenciales Evocados con el objetivo último de poder realizar un seguimiento evolutivo que muestre de forma objetiva la posible mejora tras la administración de un tratamiento de rehabilitación. De forma alternativa, y con la intención de poder realizar estudios posteriores que muestren las posibles correlaciones entre los resultados neurofisiológicos y los datos conducturales obtenidos tras la realización de tareas cognitivas, el sistema va a permitir almacenar datos referentes diferentes tareas que se utilizan en el proceso de evaluación y rehabilitación de funciones cognitivas como es la Memoria Operativa (Tarea tipo *Sternberg*, Tarea *2-Back* y Tarea *Go-NoGo*) y la Atención (Tarea *Attention-Network* y Tarea de tipo *Stroop*).

4

Diseño e Implementación

4.1 Clases de diseño.

Una vez terminada la fase de análisis y concienciados del diseño que seguirá nuestra aplicación, estableceremos la arquitectura del sistema. Nuestra aplicación se basa en un sistema simple diseñado para funcionar en una sola máquina, consistente en una arquitectura cliente/servidor.

Mediante el diagrama de clases en la fase de diseño representaremos la estructura estática del sistema, incluyendo una colección de elementos de modelado estáticos, tales como clases y relaciones. Los pasos seguidos para la realización de este diagrama de clases han sido:

1. Identificar las clases del software (las obtendremos estudiando todos los diagramas de casos de uso).
2. Incorporación de Asociaciones (agregación y composición) y Navegabilidad. La Navegabilidad indica la posibilidad de navegar unidireccionalmente en una asociación, desde los objetos fuente hasta la clase destino. También significa visibilidad (generalmente de atributos). Para nuestra aplicación. La mayoría de las asociaciones serán de composición, ya que las clases agregadas dependen totalmente de la existencia de sus clases “padre”.
3. Definir la herencia, donde los atributos y operaciones de la clase más general (superclase) son atributos de la clase especializada (subclase).

En la Figura 4.1 se muestra la estructura del Diagrama de Clases de la fase de diseño.

A continuación se mostrarán las funciones y atributos de algunas de las clases más significativas del sistema. Al ser una aplicación diseñada en PHP + JavaScript, la funcionalidad del sistema se ha visto dividida entre estos lenguajes, aunque las clases repre-

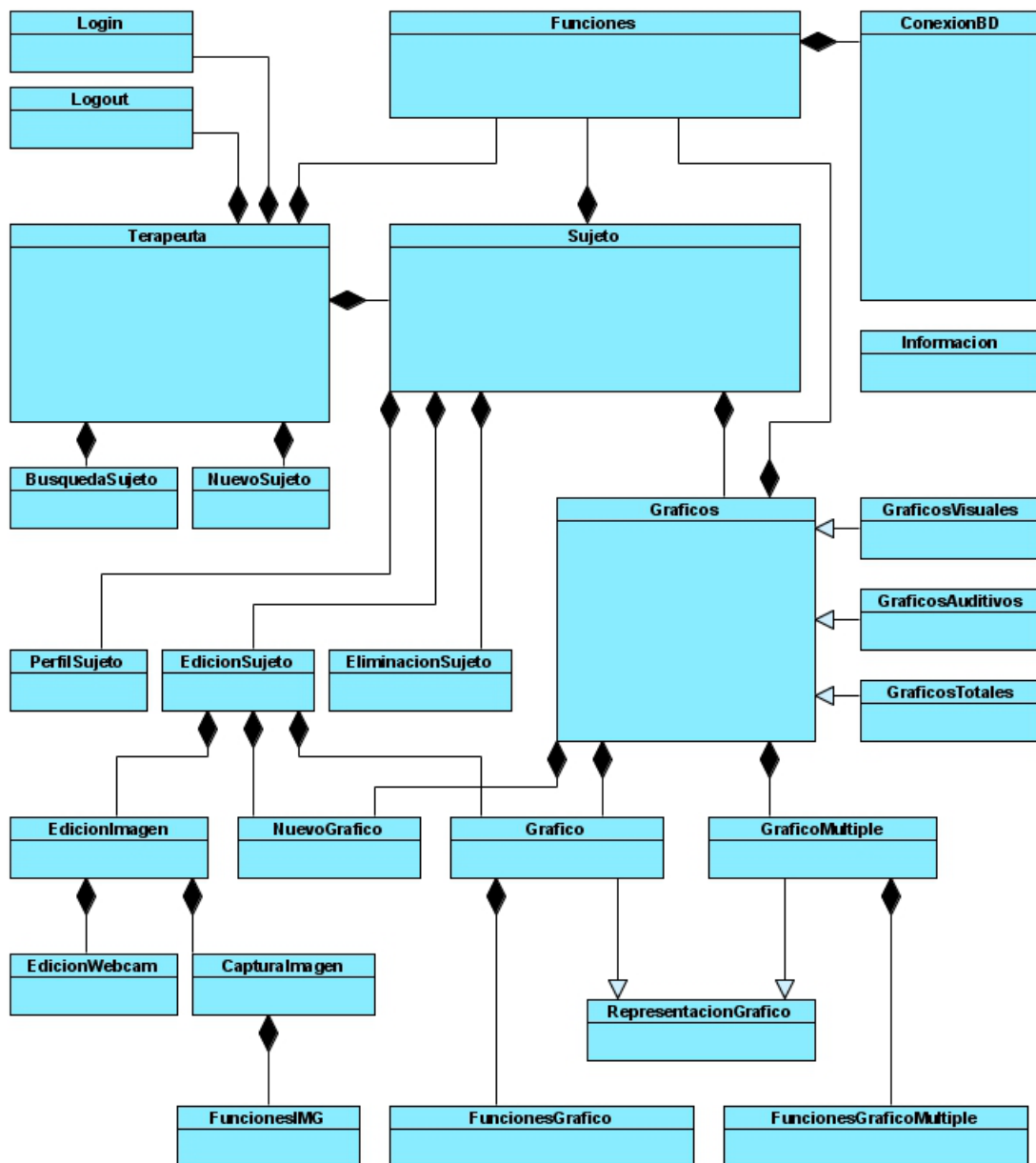


Figura 4.1: Diagrama de Clases de la fase de diseño.

sentadas a continuación muestran únicamente los atributos y métodos PHP.

- *Clase ConexionBD* (Figura 4.2). Esta clase es la encargada de conectar con la base de datos e interactuar con ella. Todas las consultas, tanto de inserción, eliminación, actualización u obtención de datos, se encuentran definidas aquí. Además, dispone del método `abrir_conexion()` que es el encargado de establecer la conexión con el servidor que contiene la base de datos (mediante un username, un password y un nombre de host) y además selecciona la base de datos con la que deseamos trabajar.
- *Clase Funciones* (Figura 4.3). Esta clase contiene todas las funciones generales que usan las demás clases principales de la aplicación. Algunas de ellas se comunican con

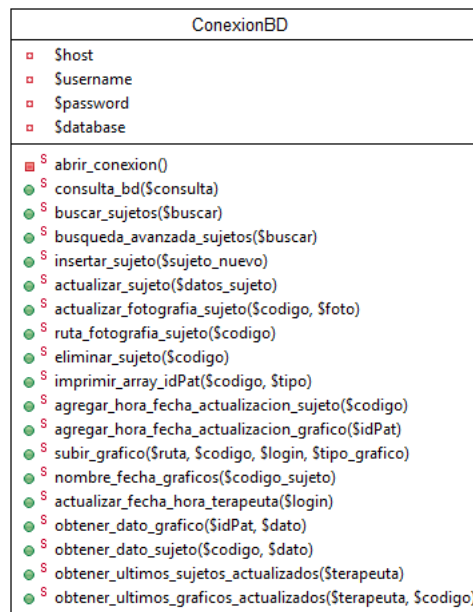


Figura 4.2: Clase ConexionBD.

la base de datos mediante la clase *ConexionBD* que hemos comentado anteriormente. Esta clase no dispone de atributos, ya que se usa como biblioteca de funciones estática.



Figura 4.3: Clase funciones.

- *Clase Grafico* (Figura 4.4). Esta clase contiene los atributos básicos de un gráfico para nuestra aplicación. Estos datos se obtienen haciendo uso de la conexión con la base de datos, aunque la mayoría de funciones utilizadas para el uso y gestión de gráficos en la aplicación se encuentran implementadas en JavaScript.
- *Clases Login y Logout* (Figura 4.5). Estas clases son las encargadas de iniciar y terminar la sesión del terapeuta. Sus atributos son los necesarios para poder establecer

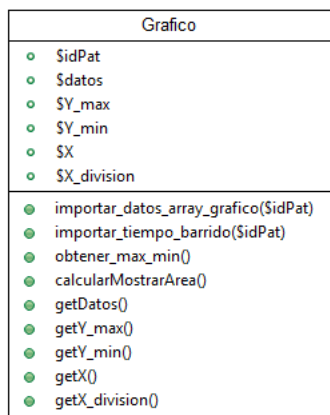


Figura 4.4: Clase Grafico.

una conexión con la base de datos y registrarse como usuario de la aplicación. Las funciones que implementan tienen como propósito principal la comprobación de identidad así como el registro de sesiones en la base de datos.

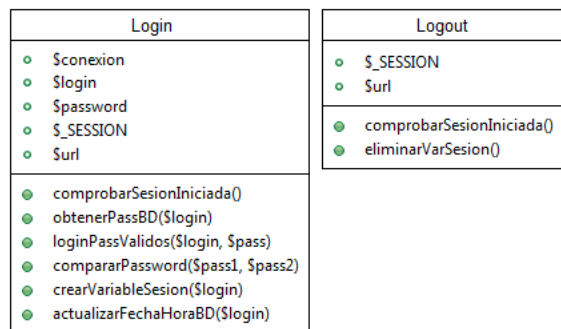


Figura 4.5: Clases Login y Logout.

4.2 Diagramas de Secuencia.

A continuación se mostrarán una serie de diagramas de secuencia que representan los casos de uso más representativos de la aplicación. Estos diagramas reflejan la interacción de un conjunto de objetos en la aplicación a través del tiempo. Mientras que el diagrama de casos de uso permite el modelado de una vista general del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario así como los mensajes intercambiados entre los distintos objetos. Se mostrarán los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

Iniciar Sesión. (Figura 4.6)

Cuando el terapeuta desea iniciar una nueva sesión en la aplicación, lo hace desde la página principal, la cual manda la orden de login a la clase “Login” que, a su vez, realiza una serie de operaciones que comentamos a continuación. En primer lugar, comprueba si ya hay una sesión iniciada por este usuario. De no ser así, comprueba que los campos de login y password especificados por el usuario son válidos (no tienen caracteres extraños o cadenas vacías). Una vez realizadas esta serie de comprobaciones, se lanza una consulta a la base de datos a través de la clase “ConexionBD”. Primero se realizará la conexión con la misma y, una vez verificada, se consultará el password asociado al terapeuta indicado. A continuación, la clase Login comparará el password introducido por el usuario con el recibido por la consulta a la base de datos. Si las contraseñas son idénticas se creará una variable de sesión y se volverá a lanzar una consulta a la base de datos para introducir la fecha y hora de inicio de sesión del terapeuta en la aplicación.

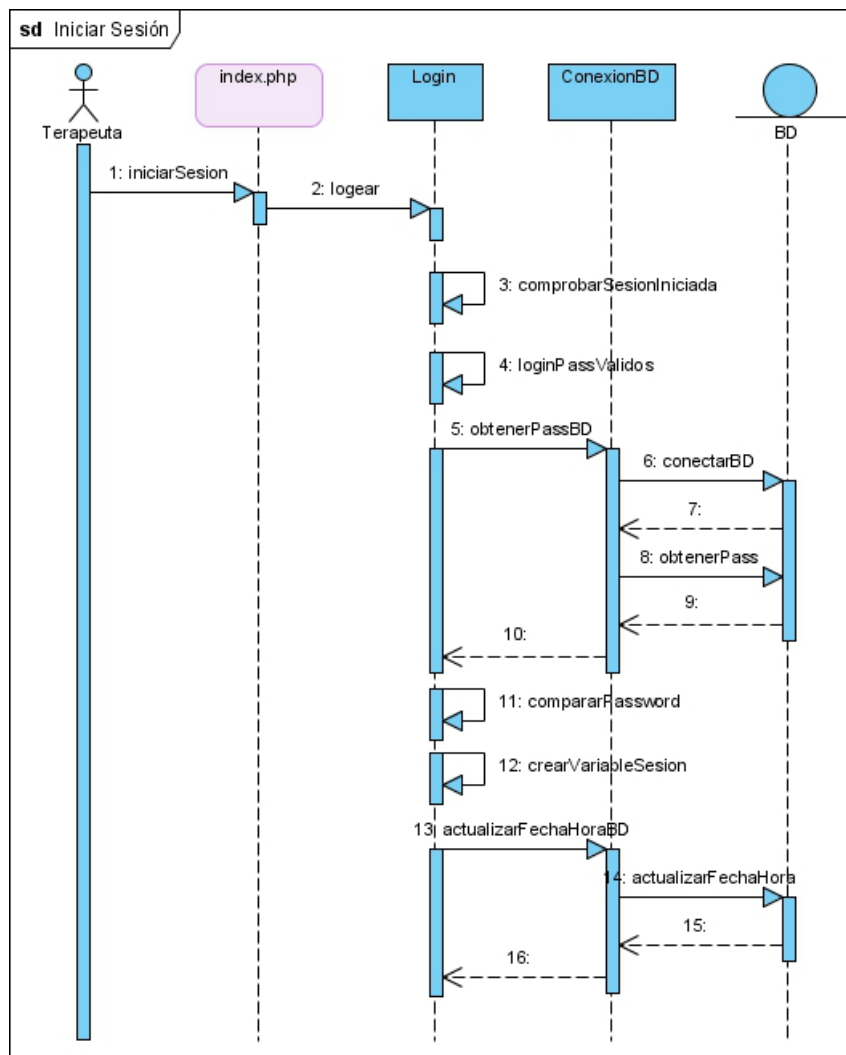


Figura 4.6: Diagrama de Secuencia “Iniciar Sesión”.

Terminar Sesión. (Figura 4.7)

Si el terapeuta ha finalizado el uso de la aplicación y desea terminar su sesión, lo hará de la siguiente forma. Desde la página principal de la aplicación indicará que desea terminar sesión. Automáticamente, se lanzará un objeto de tipo Logout, el cual realizará dos acciones: comprobar si realmente hay una sesión iniciada para terminar y, de ser así, borrar la variable de sesión existente. Una vez hecho esto, se devolverá la información del fin de sesión a la página principal de la aplicación.

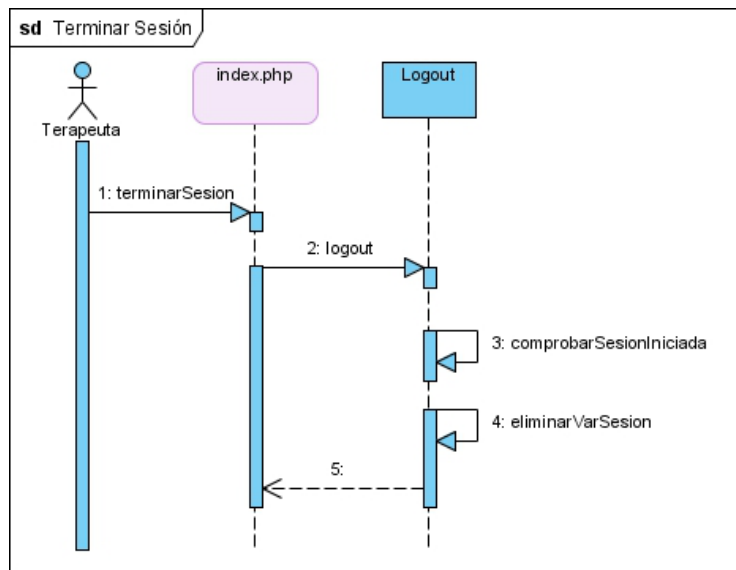


Figura 4.7: Diagrama de Secuencia “Terminar Sesión”.

Eliminar Sujeto. (Figura 4.8)

El terapeuta podrá eliminar un determinado sujeto. Para ello, primero deberá acceder a un listado donde se mostrarán todos los sujetos que tiene asignados. Estos se obtendrán de una consulta a la base de datos mediante la clase *ConexionBD*, la cual devolverá todos los sujetos que fueron dados de alta por el terapeuta que inició la sesión. Una vez obtenido el listado, se mostrará cada elemento en una tabla mediante un bucle. A partir de este momento es cuando se podrá realizar la acción de eliminar a uno de los sujetos mostrados, solicitando al terapeuta confirmación de eliminación para evitar acciones no deseadas. Una vez obtenida esta confirmación se procederá a la eliminación en la base de datos, tanto del sujeto como de todos los gráficos pertenecientes a éste.

Registrar Nuevo Sujeto. (Figura 4.9)

El terapeuta podrá dar de alta nuevos sujetos en la aplicación de la siguiente forma. Desde la página terapeuta se indicará la intención de crear un nuevo sujeto asociado al terapeuta que ha iniciado sesión. Mediante la clase *NuevoSujeto* se introducirán los datos necesarios para registrar un nuevo Sujeto en la base de datos (nombre, apellidos, código, edad, fecha de nacimiento, sexo, nivel educativo, mano dominante, visión, antecedentes neurológicos, antecedentes psiquiátricos y observaciones), aunque el único campo obliga-

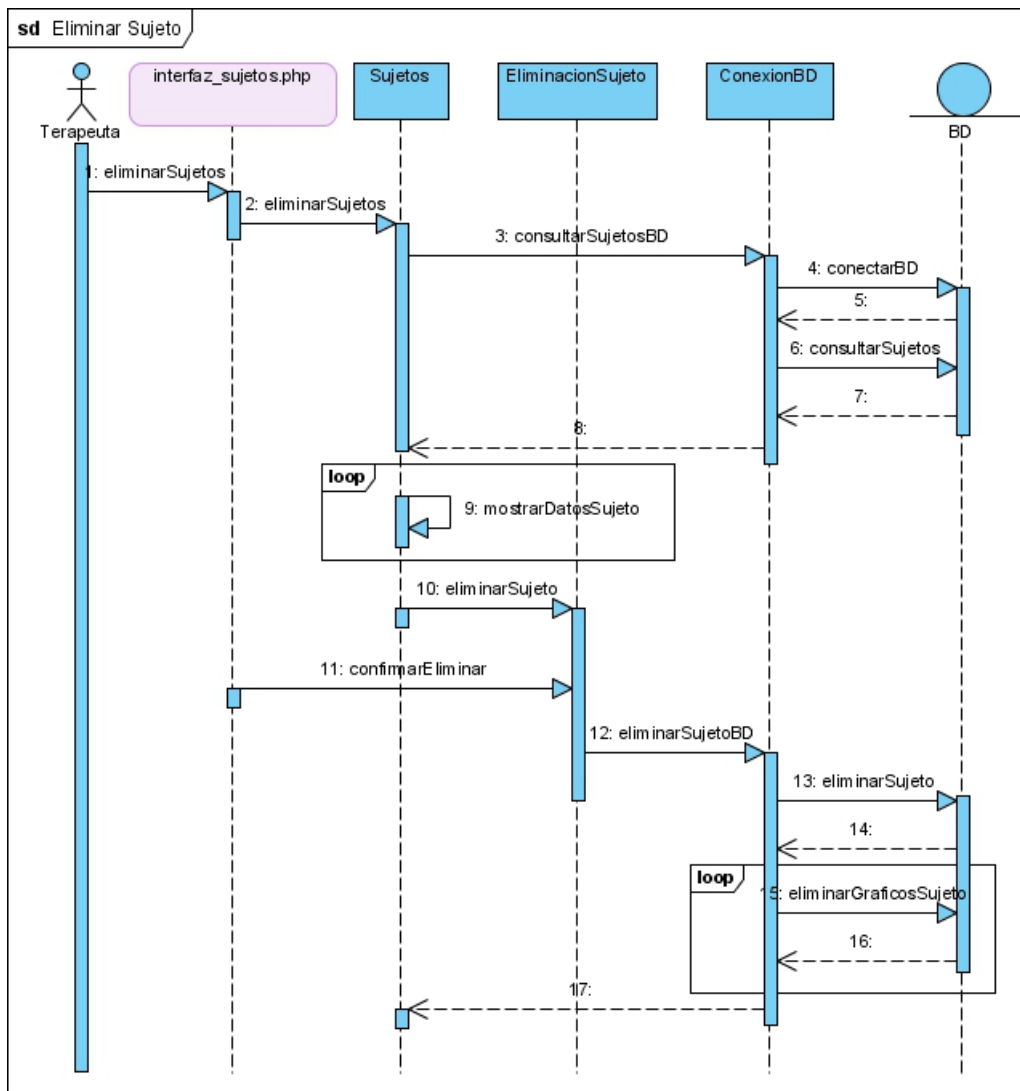


Figura 4.8: Diagrama de Secuencia “Eliminar Sujeto”.

torio es el de código. Una vez introducidos todos los datos, se realizará una consulta a la base de datos que devolverá todos los códigos de los sujetos pertenecientes al terapeuta que inició sesión. Estos códigos se compararán con el introducido para el nuevo sujeto y, si alguno de ellos es exactamente igual, se informará al terapeuta para que se vuelva a introducir un código único. Tras esto, se procederá a registrar el nuevo sujeto en la base de datos (mediante la clase *ConexionBD* nuevamente).

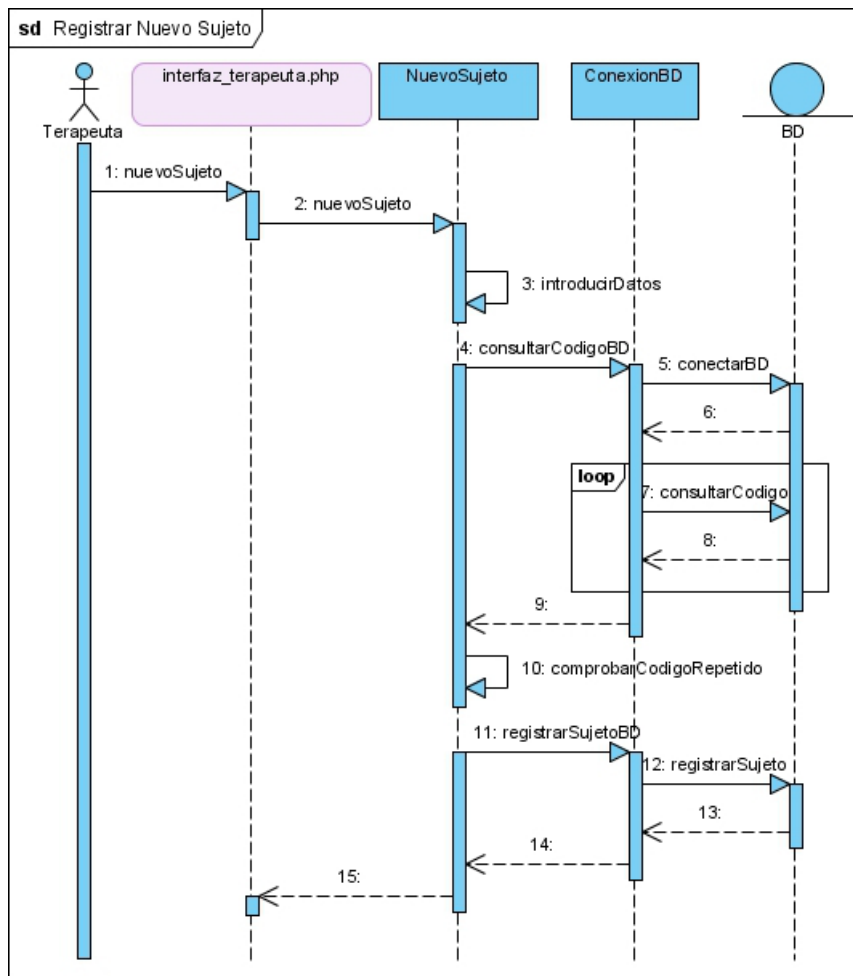


Figura 4.9: Diagrama de Secuencia “Registrar Nuevo Sujeto”.

Actualizar Sujeto. (Figura 4.10)

La forma de actualizar un sujeto es, en parte, similar a la usada para registrar un nuevo sujeto en la aplicación. Primero, el terapeuta deberá acceder a un listado donde se mostrarán todos los sujetos que tiene asignados. Este listado se obtendrá a partir del resultado de una consulta a la base de datos mediante la clase *ConexionBD*, la cual devolverá todos los sujetos que fueron dados de alta por el terapeuta que inició sesión. Una vez obtenido el resultado, se mostrarán al usuario en una tabla de forma iterativa. Tras esto, se procederá a la actualización del sujeto seleccionado mediante la clase *EdicionSujeto*, en la cual se editarán los campos que se deseen, presentando la opción de modificar la imagen asignada al sujeto. Esto se hará mediante la clase *EdicionImagen*, con la que se podrá seleccionar un archivo de imagen fuente, permitiendo la selección de las dimensiones óptimas para la aplicación. Una vez actualizados los datos y cambiada la imagen (si se desea) se procederá a la actualización del sujeto en la base de datos a través la clase *ConexionBD*.

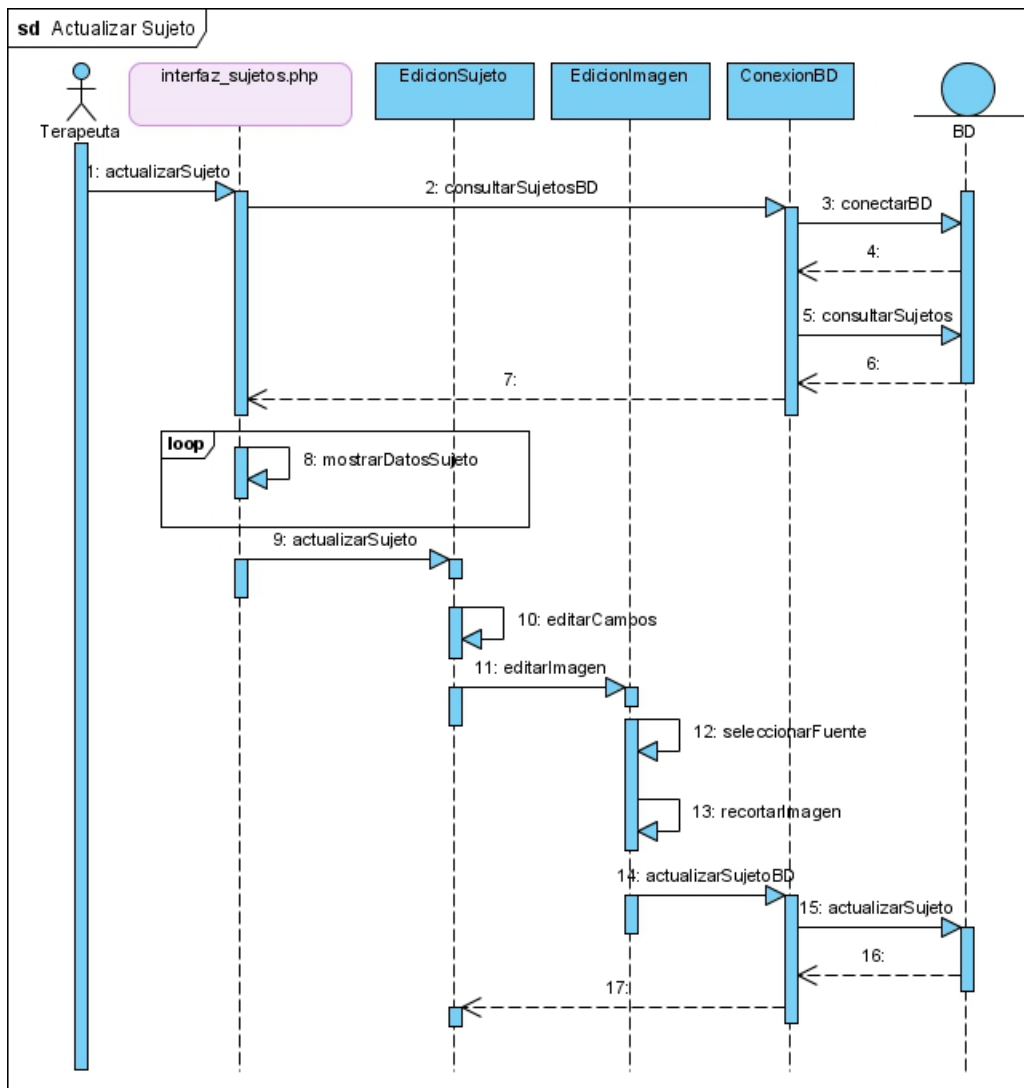


Figura 4.10: Diagrama de Secuencia “Actualizar Sujeto”.

Combinar Gráficos. (Figura 4.11)

El terapeuta tiene la opción de combinar distintos gráficos de un mismo sujeto para poder así comparar los resultados de distintas pruebas (p.e. comparar el potencial auditivo obtenido del oído izquierdo con el derecho) o realizar un seguimiento evolutivo de las pruebas cognitivas realizadas a un paciente. Para realizar esta acción, primero se debe acceder a un listado donde se mostrarán todos los sujetos que tiene asignados siguiendo el mismo procedimiento comentado en la sección anterior. A continuación, el terapeuta seleccionará uno de los sujetos recuperados de la consulta y realizará la acción de mostrar gráficos. Mediante la clase *Gráficos* se lanzará una nueva consulta que obtendrá todos los gráficos (o únicamente los visuales o auditivos) pertenecientes al sujeto seleccionado. Estos gráficos se representarán mediante un estructura iterativa que recorrerá la respuesta obtenida de la base de datos. A continuación, el terapeuta seleccionará todos y cada uno de los gráficos que desea combinar y, tras esta acción, se creará un nuevo objeto de la clase *GraficoMultiple*, el cual lanzará una serie de consultas a la base de datos para obtener

todos los datos necesarios para la visualización de cada uno de los gráficos seleccionados por el terapeuta. Una vez obtenidos la información necesaria, ésta se mostrará mediante la función *mostrarGráficosCombinados()*.

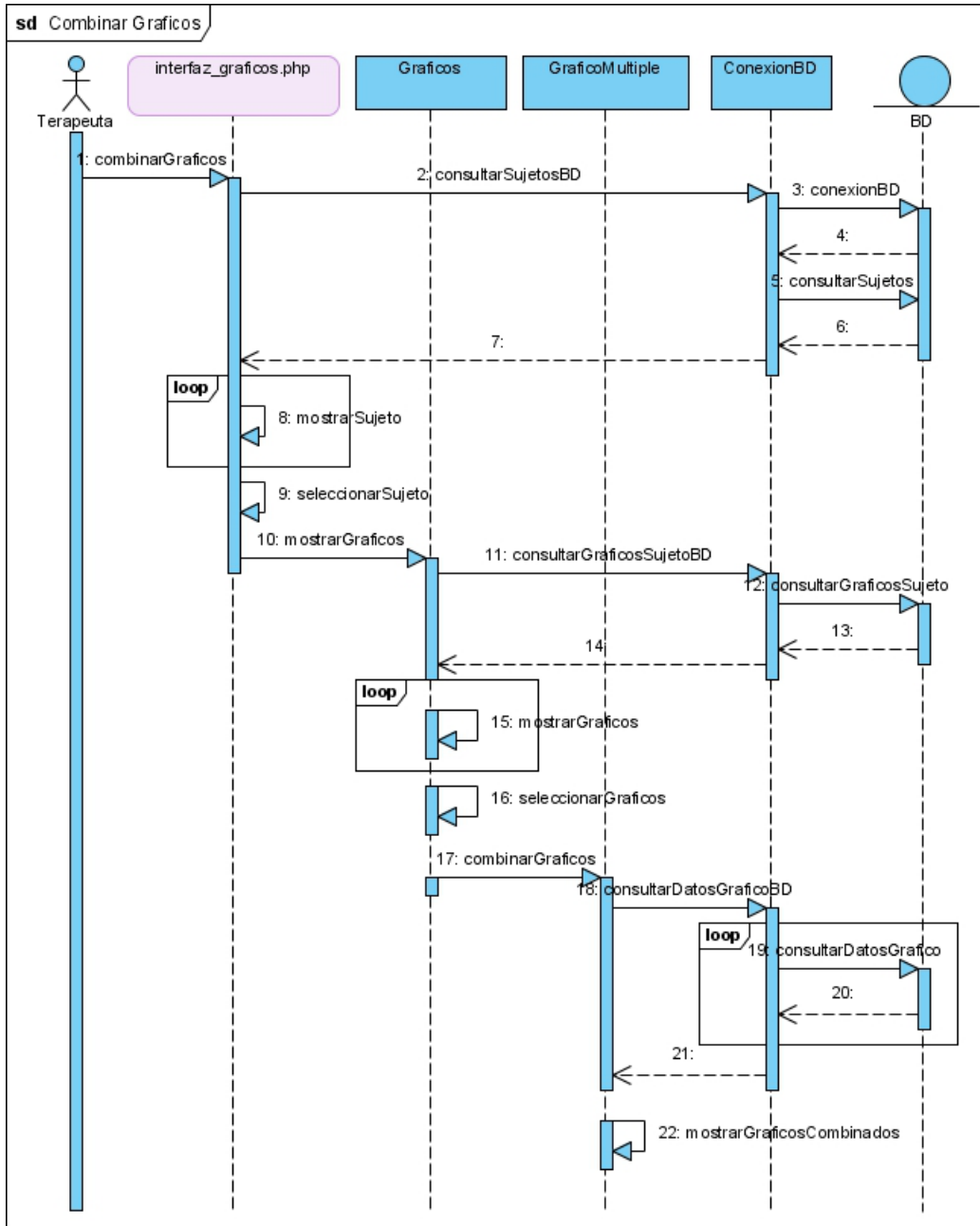


Figura 4.11: Diagrama de Secuencia “Combinar Gráficos”.

Cargar gráfico. (Figura 4.12)

Para cargar un gráfico perteneciente a un sujeto particular desde la aplicación se seguirán los siguientes pasos. Para comenzar, se deberá acceder a un listado donde se mostrarán todos los sujetos que el terapeuta tiene asignados siguiendo el proceso ya comentado. A continuación, deberemos editar el sujeto seleccionado mediante la clase *EdicionSujeto*, con la que podremos elegir el tipo de gráfico que deseamos cargar (visual o auditivo) y, a continuación, seleccionar un archivo de datos exportado de la aplicación *MaxG EP-01*. Una vez realizado este proceso, cargaremos el gráfico con la clase *NuevoGráfico*, la cual guardará el archivo de texto en el servidor y realizará las consultas pertinentes para introducir un nuevo gráfico perteneciente al sujeto en cuestión.

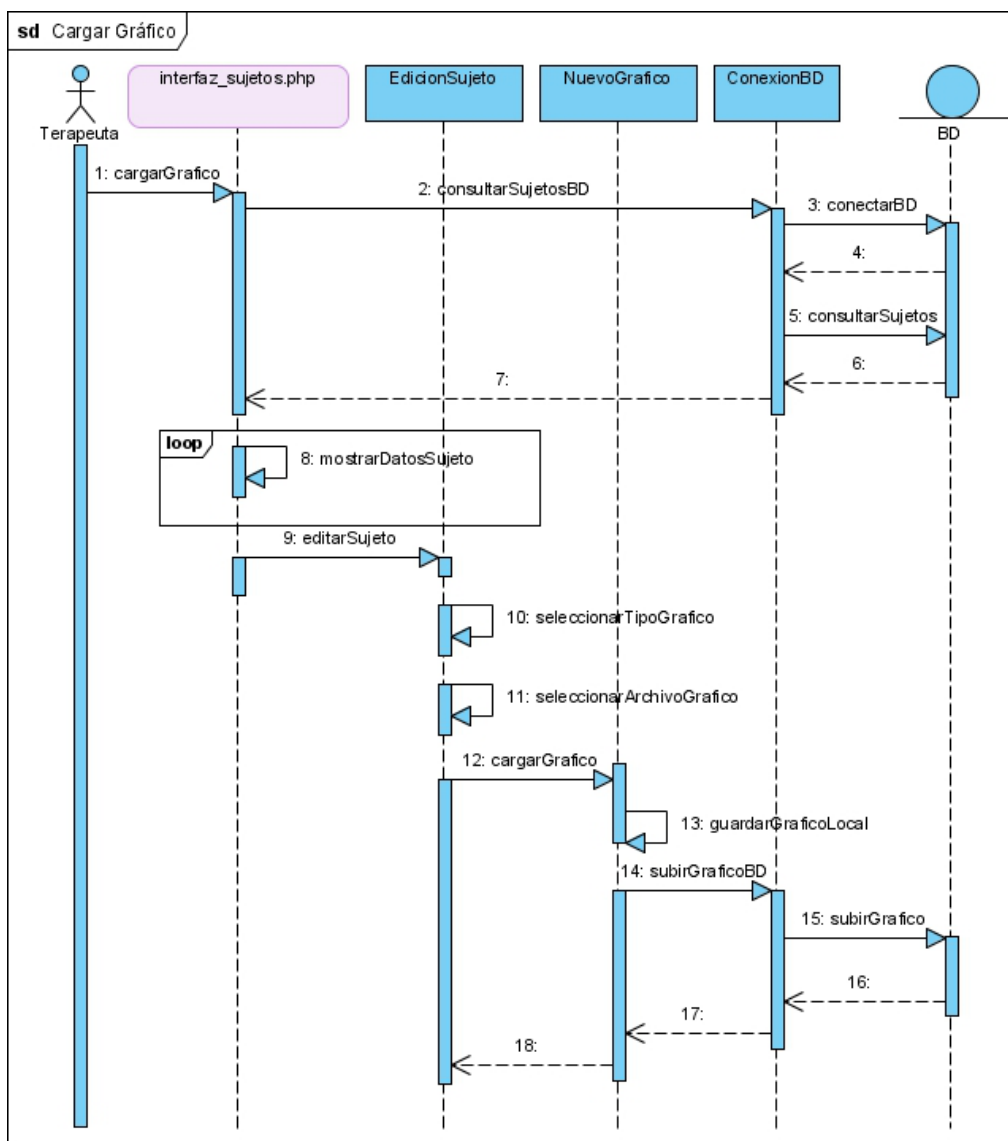


Figura 4.12: Diagrama de Secuencia “Cargar Gráfico”.

Editar gráfico simple. (Figura 4.13)

El terapeuta podrá editar gráficos de forma simple o múltiple (si se han combinado previamente). Para editar un gráfico simple se seguirán los siguientes pasos. Primero se deberá acceder a un listado donde se mostrarán todos los sujetos que el terapeuta tiene asignados. A continuación, el terapeuta seleccionará uno de los sujetos recuperados de la consulta y realizará la acción de mostrar gráficos. Mediante la clase *Gráficos* se lanzará una nueva consulta que obtendrá todos los gráficos (o únicamente los visuales o auditivos) pertenecientes al sujeto seleccionado. Estos gráficos se representarán mediante un bucle que recorrerá la respuesta obtenida por la base de datos. Una vez seleccionado el gráfico que se desea editar, el terapeuta podrá pasar al panel de edición mediante la clase *GraficoSimple*, la cual consultará en la base de datos todos los parámetros del gráfico necesarios para su representación (vector de datos, amplitud máxima, tiempo,...). Una vez que la clase *GraficoSimple* dispone de todos los datos necesarios, se representará el gráfico deseado para que el terapeuta pueda comenzar con su edición.

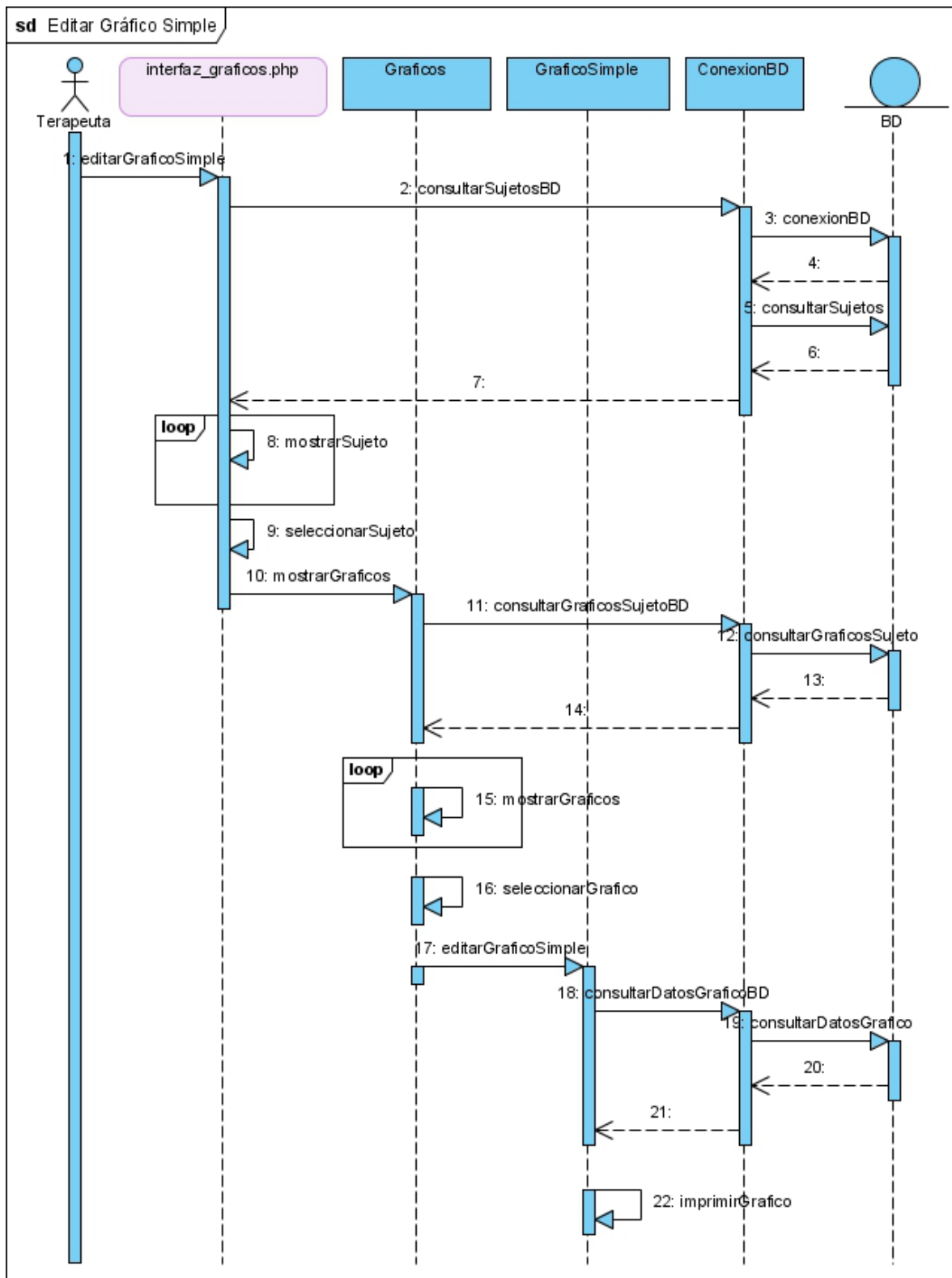


Figura 4.13: Diagrama de Secuencia “Editar Gráfico Simple”.

4.3 Diseño de la Interfaz de usuario.

En esta sección se mostrará la estructura de la interfaz de usuario (Figura 4.14). Esta estructura mostrará únicamente los ficheros *.php* de los que consta la aplicación y con los que el usuario podrá interactuar indicando, en cada caso, los enlaces entre ellos. Por este motivo, las hojas de estilo (*.css*), los scripts (*.js*) e incluso las bibliotecas de funciones *php* no se mostrarán en el siguiente diagrama.

Los enlaces mostrados en el diagrama son unidireccionales, es decir, cada página de la aplicación consta de un enlace a su página “hija” y ésta, a su vez, otro enlace a su página “padre”, pero en este caso no se ha ilustrado por motivos de legibilidad.

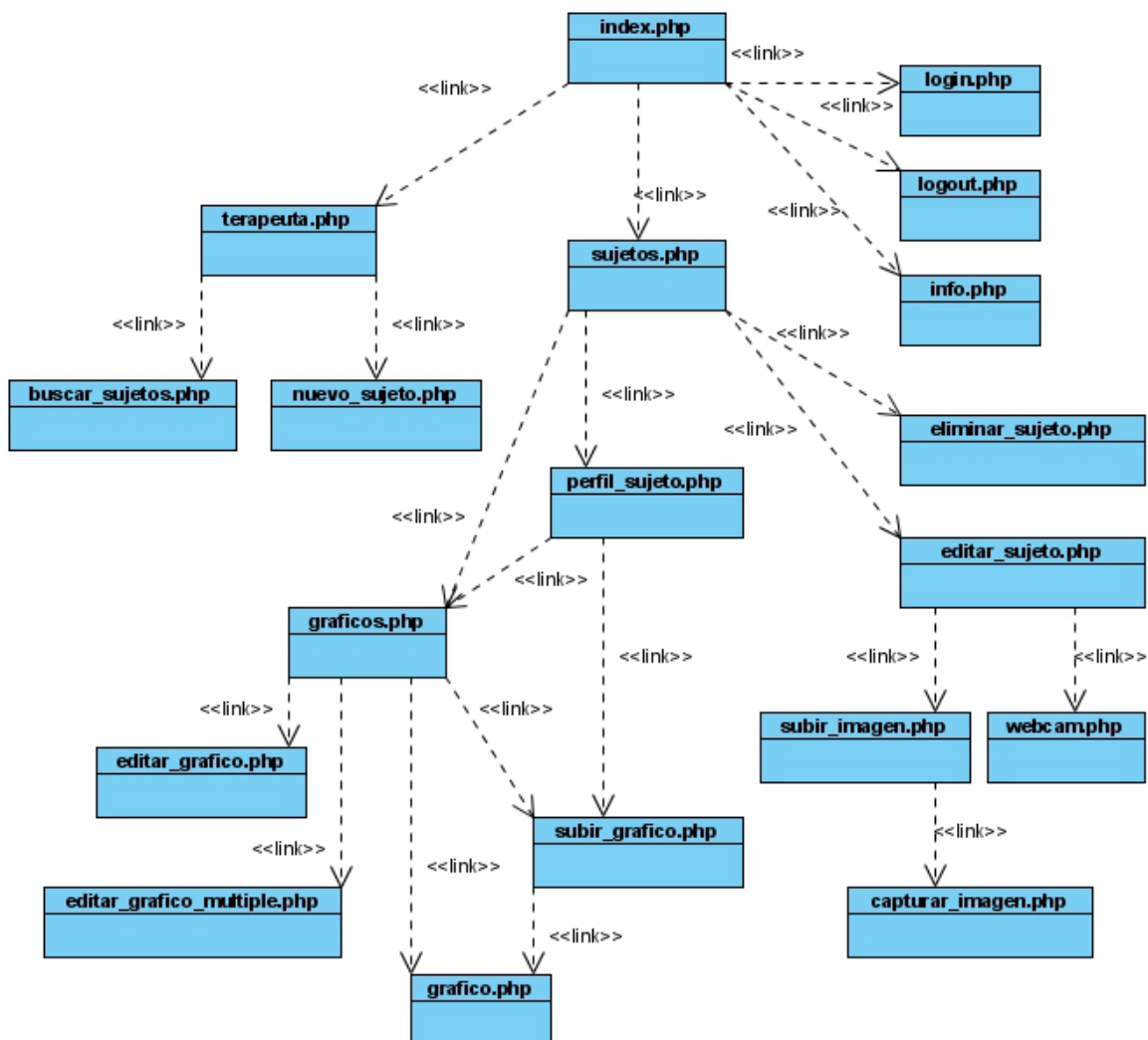


Figura 4.14: Interfaz de usuario de *AWARD Potentials*.

4.4 Diseño de la base de datos.

La base de datos de la aplicación está compuesta por un total de ocho tablas. Las tres principales son las encargadas de guardar los registros pertenecientes a los Terapeutas, Sujetos y Gráficos. Las otras cinco guardan información de una serie de tareas encargadas de la evaluación de las funciones ejecutivas y la atención en el TDAH.

4.4.1. Base de datos de *AWARD Potentials*.

A continuación se describirán las tablas que componen la base de datos de la aplicación, así como los campos (atributos) que componen cada una de ellas.

- En la tabla **Terapeuta** (4.1) cada registro pertenece a un actor Terapeuta. El campo clave de cada registro será el *login* (elegido por cada terapeuta a la hora de registrarse y único en todos los casos). Además, se guardará información útil para el inicio de sesiones (*password*), información personal del terapeuta (nombre y apellidos), el tipo de terapeuta (campo que indica si el terapeuta es administrador o no y, por lo tanto, si puede ejecutar funciones de aceptación/cancelación de nuevos terapeutas). Y por último, una serie de campos que indican la fecha y hora de la sesión iniciada por el terapeuta.

Dato	Campo	Tipo	Descripción
Identificador del terapeuta	login	varchar(20)	
Password del terapeuta	password	varchar(20)	
Nombre	nombre	varchar(120)	
Apellidos	apellidos	varchar(120)	
Tipo de terapeuta	tipo	varchar(1)	Tipo 1 (administrador) ó 2 (normal)
Hora de la última visita	hora_ult_visita	varchar(5)	
Fecha de la última visita	fecha_ult_visita	varchar(10)	

Tabla 4.1: Tabla Terapeuta de la Base de datos.

- En la tabla **Sujeto** (4.2) la mayoría de los campos son de carácter personal (nombre, apellidos, ...). El identificador es un código único para cada sujeto (se puede utilizar el DNI) que el terapeuta deberá elegir y que, en caso de ser necesario, la aplicación informará al terapeuta de si está repetido o no de forma eficiente mediante la utilización de conexiones asíncronas que permite AJAX. Además, se guardan una serie de datos sobre las características cognitivas del sujeto y sus antecedentes neurológicos y psiquiátricos. También se guarda información sobre la fecha y hora de la última modificación que se hizo para un sujeto determinado (esto puede ser, subir un gráfico, cambiar el perfil, introducir la fotografía del sujeto en el sistema, etc). Del mismo modo, otro de los campos es la ruta de la imagen del sujeto en el sistema y, por último, el identificador del terapeuta que dió de alta a dicho sujeto.

Dato	Campo	Tipo	Descripción
Nombre	nombre	varchar(120)	
Apellidos	apellidos	varchar(120)	
Código	codigo	int(11)	
Edad	edad	int(11)	
Fecha de nacimiento	fecha_nacimiento	varchar(10)	
Sexo	sexo	varchar(10)	
Nivel educativo	nivel_educativo	int(11)	Cuatro opciones: 1/2/3/4
Mano dominante	mano_dominante	varchar(15)	Tres opciones: Derecha, Izquierda o Ambidiestro
Visión	vision	varchar(10)	Dos opciones: Normal o Corregida
Antecedentes neurológicos	antecedentes_neurologicos	varchar(1500)	
Antecedentes psiquiátricos	antecedentes_psiquiatricos	varchar(1500)	
Observaciones	observaciones	varchar(1500)	
Hora de la última actualización	hora_ult_actualizacion	varchar(10)	
Fecha de la última actualización	fecha_ult_actualizacion	varchar(10)	
Variable de actualización	var_actualizacion	bigint(14)	Variable entera basada en la fecha y hora de actualización
Imagen del sujeto	imagen	varchar(120)	Ruta de la imagen en disco
Identificador del Terapeuta	login	varchar(20)	Login del terapeuta que dió de alta al sujeto

Tabla 4.2: Tabla Sujeto de la Base de datos.

- En la tabla **Gráfico** (4.3) se muestran los datos relativos a los gráficos procedentes de los test exportados de la aplicación *MaxG EP - 01*. La mayoría de los datos son de carácter técnico y algunos de ellos son ignorados por la aplicación, aunque se decidió mantenerlos almacenados de forma íntegra para así poder ser utilizados en futuros trabajos de *AWARD Potentials*. Algunos datos interesantes son: el identificador del test (*idPat*, utilizado como clave primaria por la aplicación), la fecha de exportación del gráfico, el nombre del gráfico, el tiempo de barrido de la señal exportada y, por supuesto, el vector de la secuencia adquirida que se representará en la aplicación. Además de estos campos, también se almacenarán otros necesarios para la interacción del gráfico con los demás elementos de la aplicación. Estos campos son: el código de sujeto al que pertenece el gráfico y el identificador del terapeuta que subió el gráfico a la aplicación.
- En la tabla **Tarea Attention Network Test** (4.4) se muestran los datos de esta tarea cognitiva que se ha desarrollado para medir la función de las tres diferentes

Dato	Campo	Tipo	Descripción
Código del sujeto	codigo_sujeto	int(11)	identificador del sujeto al que pertenece el gráfico
Fecha de exportación	fecha_exportacion	varchar(19)	Fecha en la que se exportó el gráfico
Identificador del gráfico	idPat	varchar(20)	
Nombre	nombrePat	varchar(80)	
Tipo de test	tipoTest	int(1)	Siempre 0
Etiqueta del gráfico	etiqueta	varchar(100)	Nombre del gráfico
Número de la traza	numTraza	int(11)	Número de la traza que se exportó, de 0 a 31
Tiempo de barrido	tSweep	int(11)	Tiempo de barrido de señal, en milisegundos
Índice de tiempo	dura	int(11)	Índice del TSWEEP en el combo box de la aplicación
Número de barridos	nSweeps	int(11)	Número de barridos sumados para conseguir el promedio
Barridos rechazados	nRejects	int(11)	Número de barridos rechazados para el promedio (da idea de la calidad del proceso de registro)
Ganancia total	gt	int(11)	Ganancia total del canal (factor multiplicativo)
Vpaso	vpaso	int(11)	Factor interno
Etiqueta 1	labelM1	int(11)	Etiqueta de la marca 1
Número de muestra 1	numSampleM1	int(11)	Número de muestra donde se ha colocado la marca 1
Tiempo desde estímulo 1	timeM1	int(11)	Tiempo desde estímulo para la marca 1 (M1), en milisegundos
Amplitud 1	ampM1	int(11)	Amplitud de la muestra en μV
...			
Vector de enteros	intVect	varchar(7000)	Muestras de la secuencia adquirida, array[0..999] de enteros
Identificador terapeuta	login	varchar(20)	Identificador del terapeuta que subió este gráfico

Tabla 4.3: Tabla Gráfico de la Base de datos.

redes atencionales: alerta, orientación y control ejecutivo. Esta tarea se basa en la idea de que los aspectos de la atención se distinguen y que los individuos pueden diferir en las funciones atencionales por separado. Por este motivo, en la base de datos se guardarán las puntuaciones y errores de cada una de las redes atencionales por separado.

- En la tabla **Tarea Stroop** (4.5) se muestran los datos para esta tarea que es una de las más clásicas para evaluar la Función Ejecutiva en niños y adultos. Consta de tres

Dato	Campo	Tipo	Descripción
Puntuación de Alerta	alerta	varchar(10)	Dato en milisegundos
Errores de Alerta	e_alerta	varchar(10)	Porcentaje de errores
Puntuación de Orientación	orientacion	varchar(10)	Dato en milisegundos
Errores de Orientación	e_orientacion	varchar(10)	Porcentaje de errores
Puntuación de Control Ejecutivo	control	varchar(10)	Dato en milisegundos
Errores de Control Ejecutivo	e_control	varchar(10)	Porcentaje de errores

Tabla 4.4: Tabla *Attention Network Test* de la Base de datos.

partes: la primera consiste en la denominación del color en que están dibujadas unas figuras. La segunda parte consiste en la lectura de palabras (nombres de colores), que están escritas en color negro. La tercera parte, denominada tarea de interferencia, consiste en la lectura de una lista de palabras (nombres de colores), que están impresas en un color diferente, sin que exista coincidencia entre el nombre del color y el propio color con el que está impreso. Así, si aparece escrita la palabra “rojo”, el color con el que se muestra nunca será el rojo, sino verde, negro, o azul. La prueba no puede ser aplicada a personas con una baja competencia lectora ni a niños que no tengan cierto nivel de lectura automatizada, y sirve para medir la atención sostenida y selectiva, capacidad para inhibir y capacidad para clasificar información, reaccionando selectivamente ante dicha información. Los campos guardados en la tabla para esta tarea son los ensayos neutrales e incongruentes y sus errores. [17]

Dato	Campo	Tipo	Descripción
Ensayos neutrales	neutrales	varchar(10)	Dato en milisegundos
Errores neutrales	e_neutrales	varchar(10)	Porcentaje de errores
Ensayos incongruentes	incongruentes	varchar(10)	Dato en milisegundos
Errores incongruentes	e_incongruentes	varchar(10)	Porcentaje de errores

Tabla 4.5: Tabla *Tarea Stroop* de la Base de datos.

- En la tabla **Tarea Sternberg** (4.6) se muestran los datos de esta tarea que consiste en proponer al sujeto que memorice una serie de dígitos en su memoria de trabajo. A este sujeto se les muestra un estímulo que consiste en una cadena de dígitos del 0 al 9 (estímulo de muestra). Tras un tiempo de demora (usualmente, dos segundos), se le muestra una nueva pantalla con un único dígito (estímulo de comparación). El sujeto deberá responder lo más rápido posible indicando si el estímulo de comparación pertenece al estímulo de muestra. Sucesivamente se va aumentando el número de dígitos que aparecen en la serie. La idea es que, a medida que el tamaño del conjunto de dígitos aumenta, el número de procesos que necesitan ser completados antes de que el sujeto tome una decisión también aumenta. En esta tabla de la base de datos se guardan el conjunto estimular previo y los errores para cadenas de dígitos de entre 3 y 9 caracteres.

Dato	Campo	Tipo	Descripción
Conjunto estimular previo de longitud 3	cep3	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 3	e_cep3	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 4	cep4	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 4	e_cep4	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 5	cep5	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 5	e_cep5	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 6	cep6	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 6	e_cep6	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 7	cep7	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 7	e_cep7	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 8	cep8	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 8	e_cep8	varchar(10)	Porcentaje de errores
Conjunto estimular previo de longitud 9	cep9	varchar(10)	Dato en milisegundos
Errores del Conjunto estimular previo de longitud 9	e_cep9	varchar(10)	Porcentaje de errores

Tabla 4.6: Tabla *Tarea Sternberg* de la Base de datos.

- En la tabla **Tarea 2-back** (4.7) se muestran los datos registrados pertenecientes a la realización de esta tarea. La tarea 2-back es una tarea cognitiva que se utiliza para evaluar la memoria de trabajo. Consiste en presentar una secuencia de estímulos que el sujeto debe ir controlando para indicar cuándo el estímulo que se muestra en el instante actual coincide con el que se mostró 2 instantes de tiempo antes.

Dato	Campo	Tipo	Descripción
Número de aciertos	aciertos	varchar(10)	
Número de fallos	fallos	varchar(10)	
Número de falsas alarmas	falsas_alarmas	varchar(10)	
Número de rechazos correctos	rechazos_correctos	varchar(10)	
Número de respuestas perdidas	respuestas_perdidas	varchar(10)	

Tabla 4.7: Tabla *Tarea 2-back* de la Base de datos.

- En la tabla **Tarea Go-noGo** (4.8) se muestran los datos pertenecientes a estas pruebas de acción-inhibición que consisten en un método rápido y relativamente fiable para conocer el funcionamiento del área prefrontal. Consisten en la presentación de estímulos frente a los que el sujeto tiene que realizar una determinada acción (tarea “go”), junto con otros (tareas “no go”), ante los que debe inhibir la respuesta. Miden la capacidad para inhibir conductas y el grado de control de la impulsividad. Los niños sólo empiezan a tener éxito en este tipo de tareas a partir de los 5-6 años [17].

Dato	Campo	Tipo	Descripción
Número de aciertos	aciertos	varchar(10)	
Número de falsas alarmas	falsas_alarmas	varchar(10)	
d'	d	varchar(10)	
β	beta	varchar(10)	
Ensayos “Go”	ensayos	varchar(10)	Dato en milisegundos

Tabla 4.8: Tabla *Tarea Go-noGo* de la Base de datos.

4.4.2. Unificación con la base de datos de *AWARD Subject*.

AWARD Subject es una herramienta de Gestión de Datos Personales que permite almacenar tanto los datos personales, historia clínica, así como los relativos al contexto escolar y familiar de los usuarios de las aplicaciones web desarrolladas por el equipo del proyecto *AWARD*.

Es una herramienta de gestión de Bases de Datos Personales que actualmente cuenta con tres módulos principales:

- Gestor de usuarios del proyecto *AWARD*.
- Gestor de investigación genérico.
- Gestor clínico.

La aplicación *AWARD Potentials* se unirá a la base de datos de *AWARD Subject* para así unificar en última instancia todos los datos de estudio obtenidos para pacientes del proyecto *AWARD*. A continuación se muestra la estructura de la base de datos actual de *AWARD Subject*:

5

Aspectos destacados de Implementación

5.1 Diseño de aplicaciones web en PHP.

Para la realización de esta aplicación se ha utilizado como base el lenguaje de programación web PHP (acrónimo recursivo del inglés *PHP Hypertext Preprocessor*). Las cuatro grandes razones por las que se ha elegido este lenguaje han sido [18]:

- **Velocidad.** No sólo la velocidad de ejecución, la cual es importante, sino que además PHP no crea tiempos de demora en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo entorno Unix/Linux en el que, cuando se configura como módulo de Apache, está listo para ser utilizado. Está completamente escrito en C, por lo que se ejecuta rápidamente utilizando pocos recursos computacionales.
- **Estabilidad.** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de errores, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil encontrar alguno. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad.** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, los cuales pueden ser configurados desde el archivo “.ini”.
- **Simplicidad.** PHP permite a los programadores generar código de forma productiva en el menor tiempo posible. Usuarios con experiencia en C, C++ u otros lenguajes de programación pueden utilizar PHP rápidamente.

Otra característica a tener en cuenta es la conectividad. PHP dispone de una amplia gama de librerías a la que se le pueden agregar extensiones de forma sencilla. Esto le

permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos y XML entre otras.

Además, PHP se ejecuta en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en unas 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95, 98, NT, ME, 2000, XP, ...) y Mac. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al Sistema Operativo.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.

PHP posee muchas interfaces distintas para cada tipo de servidor. Actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI. Además puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.

PHP tiene una gran variedad de módulos en el caso de necesitar una interfaz para una librería en particular; de forma sencilla se puede crear una API para ésta. Algunas de las ya implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.

5.1.1. ¿PHP o ASP?

Inicialmente se estudió realizar la implementación de esta herramienta con alguno de estos dos lenguajes de programación web. Finalmente se escogió PHP por las siguientes razones [19]:

- Precio. Implementar una aplicación en PHP es más barato que hacerlo en ASP.
- Prestaciones. PHP nos proporciona muchas más prestaciones adicionales que ASP.
- Open Source. El software de código abierto es libre de utilizar, copiar y modificar. También es muy flexible, ya que nos permite ampliar el programa y personalizarlo a nuestras necesidades.
- PHP es independiente de la plataforma. Se ejecuta en Linux, Solaris, Windows y otros Sistemas Operativos. ASP sólo funciona con fiabilidad en los servidores web de Microsoft basados en Windows, lo que es una gran desventaja. Por ejemplo, ¿qué pasaría si quisiésemos cambiar de proveedor de alojamiento web a uno mejor que el que tenemos, pero está basado en Linux?.
- PHP es el lenguaje de programación web más popular del planeta. Es más popular que ASP, ColdFusion, Perl, o que cualquier lenguaje de programación web.
- En igualdad de condiciones, PHP es más rápido que ASP. ASP soporta múltiples lenguajes de programación. Esta arquitectura es inherentemente más lenta y consume más memoria que el modelo de PHP, ya que cada compilador de lenguaje ASP se ejecuta en su propio proceso.

- PHP es seguro. IIS Server es conocida por sus agujeros de seguridad.
- La codificación en PHP es más eficiente que en ASP. Por lo general, la misma funcionalidad se puede lograr en menos código en PHP que con ASP. Esto es porque el HTTP GET y variables POST son creadas automáticamente por PHP como variables globales, eliminando así el paso de la extracción de ellas en el objeto Request de ASP.

5.1.2. Conexión con la Base de Datos MySQL.

Para la implementación de la base de datos de la aplicación se ha utilizado el sistema de gestión de bases de datos MySQL. Los motivos por los cuales se ha escogido este SGBD han sido:

- MySQL software es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL.

Para la integración de nuestra aplicación con este software hemos utilizado las funciones comunes de intercambio de datos entre PHP y MySQL que se nos ofrecían, como pueden ser `mysql_connect()` o `mysql_query()`. Además, para usar con mayor facilidad estos recursos hemos creado una serie de funciones (métodos) que nos ayudan a interactuar con la base de datos de una forma rápida y sencilla. Estas funciones han sido:

- `abrir_conexion()`. Esta función contiene todos los datos para conectarse con MySQL y seleccionar la base de datos adecuada. Además, devuelve el error concreto en el caso en el que se produzca una excepción.

```
1 private static function abrir_conexion() {
2     // Datos de conexión con la base de datos.
3     $server = array(
4         'host' => 'localhost',
5         'username' => 'root',
```

```

6     'password' => '',
7     'database' => 'temporal'
8 );
9
10 // Crea la conexión.
11 $conexion = mysql_pconnect($server['host'], $server['username'], $server['password']↵
12     ])
13     or die("Error al realizar conexión con la bd");
14
15 // Selecciona la base de datos.
16 mysql_select_db($server['database'], $conexion)
17     or die("Error al conectar a la base de datos");
18
19 return $conexion;
20 }

```

- `consulta_bd()`. Abre la conexión con la función anterior y, además, realiza una consulta especificada como parámetro de entrada. Devuelve el resultado de la consulta SQL.

```

1 public static function consulta_bd($consulta) {
2     // Crea la conexión con la base de datos.
3     $conexion = funciones::abrir_conexion();
4
5     // Realiza la consulta.
6     $query = mysql_query($consulta, $conexion)
7         or die("Error al realizar consulta");
8
9     return $query;
10 }

```

Además de estas funciones, la aplicación dispone de otras mucho más específicas que se encargan de formar consultas SQL válidas para buscar, insertar, eliminar o modificar elementos o tablas de la base de datos consultada, y que veremos a continuación.

5.1.3. Inserción, eliminación y modificación de registros.

En primer lugar, la función `insertar_sujeto()` funciona de la siguiente forma:

- Se inicia la cadena SQL de inserción indicando la acción y la tabla donde tendrá lugar dicha operación.
- Por cada campo de la tabla de sujetos se comprueba si existe dicho campo y se introduce en la consulta.
- Se indica que, a continuación, se van a introducir los valores con el identificador `VALUES`.
- Se introducen todos los valores de los campos que existan en el sujeto nuevo y se cierra la consulta.
- Se lanza la consulta.
- Se actualiza la hora y fecha de actualización del sujeto.

```

1 public static function insertar_sujeto($sujeto_nuevo) {
2     $primero = true;
3     $consulta = "INSERT INTO temporal.sujeto (";
4
5     if (isset($sujeto_nuevo['nombre'])) {
6         $nombre = $sujeto_nuevo['nombre'];
7         if(!$primero) $consulta = $consulta . ", "; else $primero = false;
8         $consulta = $consulta . "nombre";
9     }
10    if (isset($sujeto_nuevo['apellidos'])) {
11        $apellidos = $sujeto_nuevo['apellidos'];
12        if(!$primero) $consulta = $consulta . ", "; else $primero = false;
13        $consulta = $consulta . "apellidos";
14    }
15    ...
16
17    $primero = true;
18    $consulta = $consulta . ") VALUES (";
19
20    if (isset($sujeto_nuevo['nombre'])) {
21        $nombre = $sujeto_nuevo['nombre'];
22        if(!$primero) $consulta = $consulta . ", "; else $primero = false;
23        $consulta = $consulta . "'$nombre'";
24    }
25    if (isset($sujeto_nuevo['apellidos'])) {
26        $apellidos = $sujeto_nuevo['apellidos'];
27        if(!$primero) $consulta = $consulta . ", "; else $primero = false;
28        $consulta = $consulta . "'$apellidos'";
29    }
30    ...
31
32    $consulta = $consulta . ");";
33    $query = funciones::consulta_bd($consulta)
34        or die("Error al realizar consulta");
35    funciones::agregar_hora_fecha_actualizacion_sujeto($codigo);
36    return $query;
37 }

```

Para eliminar un sujeto de la tabla únicamente se crea una consulta SQL de tipo DELETE y se le pasa el identificador del sujeto que deseamos eliminar.

```

1 public static function eliminar_sujeto($codigo) {
2     $consulta = $consulta . "DELETE FROM sujeto WHERE codigo = '$codigo'";
3     $query = funciones::consulta_bd($consulta)
4         or die("Error al realizar consulta 'Eliminar sujeto ' ");
5     return $query;
6 }

```

Para modificar un sujeto, el procedimiento es similar al de inserción:

- Se inicia la consulta SQL del tipo UPDATE.
- Se introducen los campos de la tabla sujeto modificados indicando el identificador y su valor.
- Se introduce la clausula WHERE indicando el código (identificador) del sujeto.
- Se lanza la consulta.
- Se modifica la hora y fecha de actualización.

```

1 public static function actualizar_sujeto($datos_sujeto) {
2     $primero = true;
3     $consulta = "UPDATE temporal.sujeto SET ";
4
5     if (isset($datos_sujeto['nombre'])) {
6         $nombre = $datos_sujeto['nombre'];
7         if(!$primero) $consulta = $consulta . ", "; else $primero = false;
8         $consulta = $consulta . "nombre = " . "'$nombre'";
9     }
10    if (isset($datos_sujeto['apellidos'])) {
11        $apellidos = $datos_sujeto['apellidos'];
12        if(!$primero) $consulta = $consulta . ", "; else $primero = false;
13        $consulta = $consulta . "apellidos = " . "'$apellidos'";
14    }
15    ...
16    $consulta = $consulta . " WHERE codigo = '$codigo'";
17    $query = funciones::consulta_bd($consulta)
18        or die("Error al realizar consulta");
19    funciones::agregar_hora_fecha_actualizacion_sujeto($codigo);
20    return $query;
21 }

```

5.1.4. Scripts de autenticación y sesiones.

Las sesiones, en aplicaciones web realizadas con PHP, nos sirven para almacenar información que se memorizará durante toda la visita del usuario (terapeuta) a nuestra aplicación web. Dicho de otra forma, un usuario puede ver varias páginas durante su paso por un sitio web y mediante la gestión de sesiones se pueden almacenar variables que podremos acceder en cualquiera de esas páginas.

Las sesiones permiten almacenar información, específica para cada usuario, durante toda su visita. Cada usuario que entra en un sitio web abre una sesión la cual será independiente de la sesión asociada con otros usuarios. En la sesión de un usuario se puede almacenar todo tipo de datos que se guardan en lo que se denominan *variables de sesión*.

PHP dispone de un método bastante cómodo de guardar datos en variables de sesión y de un juego de funciones para el trabajo con sesiones y variables de sesión. Para cada usuario, PHP internamente genera un identificador de sesión único, que sirve para saber las variables de sesión que pertenecen a cada usuario. Para conservar el identificador de sesión durante toda la visita de un usuario a una página, PHP almacena la variable de sesión en una cookie, o bien la propaga a través de la URL.

Para utilizar variables de sesión en una página, se inicia la sesión con la siguiente función:

```

1 session_start ()

```

Inicia una sesión para el terapeuta o continúa la sesión que pudiera tener abierta en otras páginas. Al hacer `session_start()` PHP internamente recibe el identificador de sesión almacenado en la cookie o el que se envíe a través de la URL. Si no existe tal identificador de sesión, simplemente lo crea.

Una vez inicializada la sesión, se permite a partir de ahora utilizar variables de sesión, es decir, almacenar datos para ese terapeuta, que se conserven durante toda su visita o bien recuperar datos almacenados en páginas que haya podido visitar anteriormente.

La sesión se tiene que inicializar antes de escribir cualquier texto en la página. De no hacerlo así, corremos el riesgo de recibir un error, porque al iniciar la sesión se deben leer las cookies del usuario, algo que no se puede hacer si ya se han enviado las cabeceras de HTTP.

Una vez iniciada la sesión podemos utilizar variables de sesión a través de la estructura `$_SESSION`, que es un array asociativo donde se accede a cada variable a partir de su nombre tal y como se indica a continuación:

```
1 | $_SESSION["nombre_de_variable"]
```

En nuestro caso, la variable de sesión que hemos utilizado ha sido el identificador del terapeuta, con lo que en todo momento la URL sabe qué terapeuta es el que está accediendo a los contenidos y, de este modo, mostrar la información pertinente.

Además, si el terapeuta no se ha registrado en la aplicación (no existe variable de sesión) la aplicación lo redirigirá a la página principal indicando un mensaje de error de la siguiente forma:

```
1 | session_start();
2 | if(!isset($_SESSION['sesion'])) {
3 |     $url = "Location:index.php?error=Sujetos";
4 |     header($url);
5 | }
```

La forma con la cual iniciamos la variable de sesión es mediante un script de autenticación. Este script se encarga de obtener el login y password introducidos por un terapeuta y validar esos datos, de tal forma que si son correctos, se inicializa la variable de sesión con el valor del login del terapeuta. A continuación se muestra el script PHP:

```
1 | // Si la sesión no está iniciada.
2 | if(!isset($_SESSION['sesion'])) {
3 |
4 |     // Comprobamos si existe la variable POST login.
5 |     if (isset($_POST['login'])){
6 |         // Inicializamos las variables $login y $password.
7 |         $login = $_POST['login'];
8 |         $password = $_POST['password'];
9 |     }
10 |
11 |     // Si la variable POST login está vacía mostramos un mensaje de error.
12 |     if (empty($login)) {
13 |         echo "No se ha introducido nombre de usuario.<br><br> <a href=\"./index.php\" ←
14 |             class=\"boton\">volver</a>";
15 |     } else {
16 |         // Si la variable POST login tiene contenido, pero la variable password no, ←
17 |         mostramos un mensaje de error.
18 |         if (empty($password)) {
19 |             echo "No se ha introducido contraseña.<br><br> <a href=\"./index.php\" ←
20 |                 class=\"boton\">volver</a>";
```

```

18
19 // Si las variables POST login y password tienen contenido.
20 } else {
21
22 // Creamos una consulta que nos devuelve el password real del login ←
    indicado.
23 $consulta = funciones::consulta_bd("SELECT terapeuta.password FROM ←
    terapeuta WHERE terapeuta.login = '$login'");
24 $terapeuta = mysql_fetch_object($consulta);
25
26 // Si no existe la respuesta SQL, indicamos un error de usuario inexistente←
    .
27 if (!(isset($terapeuta->password))) {
28     echo "Usuario inexistente.<br><br> <a href=\"./index.php\" class=\"←
        boton\">volver</a>";
29
30 // Si existe la respuesta SQL.
31 } else {
32
33 // Comparamos el password devuelto por la consulta con el introducido.
34 if (!(strcmp($password, $terapeuta->password))) {
35     // Password correcto. Mostramos mensaje e iniciamos la variable de ←
        sesión con el login del terapeuta.
36     echo "¡Bienvenido <b>Emilio</b>!<br><br> <a href=\"./logout.php\" ←
        class=\"boton\">logout</a>";
37     $_SESSION['sesion'] = $login;
38
39     // Actualizamos la fecha y hora de inicio de sesión para el ←
        terapeuta.
40     funciones::actualizar_fecha_hora_terapeuta($login);
41
42     // Redirigimos a la página inicial del terapeuta.
43     $url = "Location:terapeuta.php";
44     header($url);
45 } else {
46
47     // Password incorrecto. Mostramos mensaje de error.
48     echo "Password incorrecto.<br><br> <a href=\"./index.php\" class=\"←
        boton\">volver</a>";
49
50     }
51 }
52 }
53
54 // Si la sesión está iniciada se muestra un mensaje de bienvenida junto con el botón \←
    textit{logout}.
55 } else {
56     echo "¡Bienvenido <b>" . $_SESSION['sesion'] . "</b>!<br><br> <a href=\"./logout.←
        php\" class=\"boton\">logout</a>";
57 }

```

Del mismo modo, existe un script encargado de eliminar la variable de sesión siempre que el terapeuta decida no seguir usando temporalmente la aplicación, lo que corresponde al script del *logout*.

```

1 // Inicia la sesión.
2 session_start();
3
4 // Si existe la variable de sesión.
5 if (isset($_SESSION['sesion'])) {
6
7     // Elimina la variable de sesión.
8     unset($_SESSION['sesion']);
9     session_destroy();
10
11 // Redirige al index.php.
12 $url = "Location:index.php";

```



```
13     header($url);
14
15 // Si no existe la variable de sesión.
16 } else {
17
18     // Redirige al index.php.
19     $url = "Location:index.php";
20     header($url);
21 }
```

Lo único que hace este script es comprobar si existe la variable de sesión y eliminarla. Justo después redirige la URL a index.php.

5.2 La librería JSXGraph.

5.2.1. Estudio previo de la metodología usada.

Para la implementación del editor gráfico se decidió utilizar JSXGraph, una librería de JavaScript opensource que no necesita de ningún plugin adicional para su funcionamiento.

Antes de comenzar con la implementación se estudiaron distintas posibilidades en cuanto a la metodología y tecnologías de las que disponíamos a tal fin. A continuación se detallan las mismas, así como los pros y los contras de cada una de ellas.

5.2.2. ¿Por qué no Java o Flash?

Para empezar, nos debatíamos entre tres grandes tecnologías de programación gráfica para web: Java, Flash y JavaScript. Estudiándolas detenidamente, encontramos los siguientes problemas:

Java:

- Requiere la instalación de un plugin externo.
- Tiempo de inicialización
- Gran uso de memoria.
- Los applets de Java no se suelen encontrar fuera del sector educativo.
- Poca compatibilidad con tecnología móvil como iPhone, Blackberry, Android o Google Chrome OS.

Flash:

- Sigue habiendo problemas con entornos distintos a Windows.
- Requiere la instalación de un plugin externo.
- No hay integración sin fisuras.

En la siguiente tabla podemos ver una comparativa más detallada de estas tres tecnologías:

	JavaScript	Java	Flash
Compatibilidad con navegadores	Alta	Media	Media
Tiempo de inicialización	Bajo	Alto	Medio
Tamaño del ejecutable	Mediano (300Kb)	Grande (>1Mb)	Grande (>1Mb)
Ancho de banda	Bajo (50Kb)	Alto (>1Mb)	Alto
Plugin necesario	No	Sí	Sí
Comunicación App y Página web	Rápida (inmediata)	Lenta	Media
Comunicación App y Servidor web	Fácil (AJAX)	Difícil	Media
Velocidad de ejecución	Mediana	Rápida	Mediana

Tabla 5.1: Comparativa JavaScript, Java y Flash.

5.2.2.1. ¿Por qué JavaScript y SVG en lugar de Java/Flash?

JSXGraph utiliza SVG para la representación de sus gráficos. Los Gráficos Vectoriales Escalables (o SVG, acrónimo del inglés Scalable Vector Graphics), son una especificación utilizada para describir gráficos vectoriales bidimensionales, tanto estáticos como animados en formato XML [20]. Podemos ver un ejemplo de la potencia de los gráficos vectoriales en la figura 5.1.

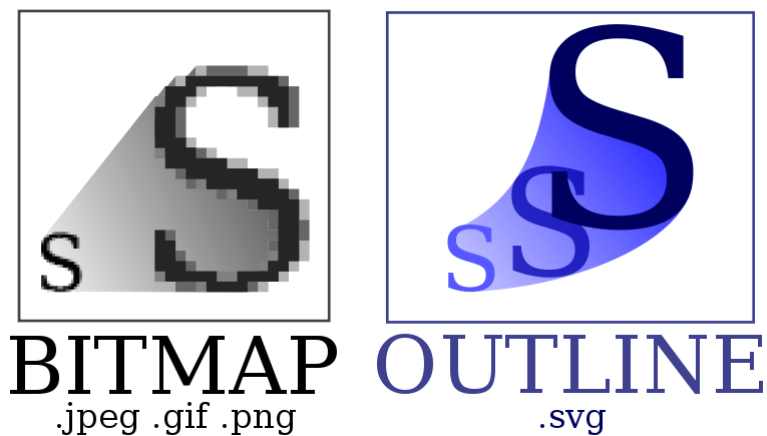


Figura 5.1: Diferencia entre imagen en mapa de bits y SVG [21].

La amplia adopción de clientes SVG, particularmente aquellos integrados nativamente en navegadores como Firefox, Opera, Konqueror o Safari, han traído un significativo cambio de imagen a la web. Una tendencia actual consiste en construir sitios web dinámicos que se comportan como aplicaciones de escritorio a través del uso de AJAX (acrónimo de Asynchronous JavaScript And XML, una técnica de desarrollo web para crear aplicaciones interactivas ricas en contenido). SVG amplía las posibilidades de AJAX, proporcionando un juego gráfico más rico a los elementos de la página, más allá de los especificados mediante HTML/CSS [22].

A continuación, en la Figura 5.2 se puede ver el proceso que sigue un gráfico creado con esta librería hasta ser renderizado y visualizado por los navegadores más habituales.

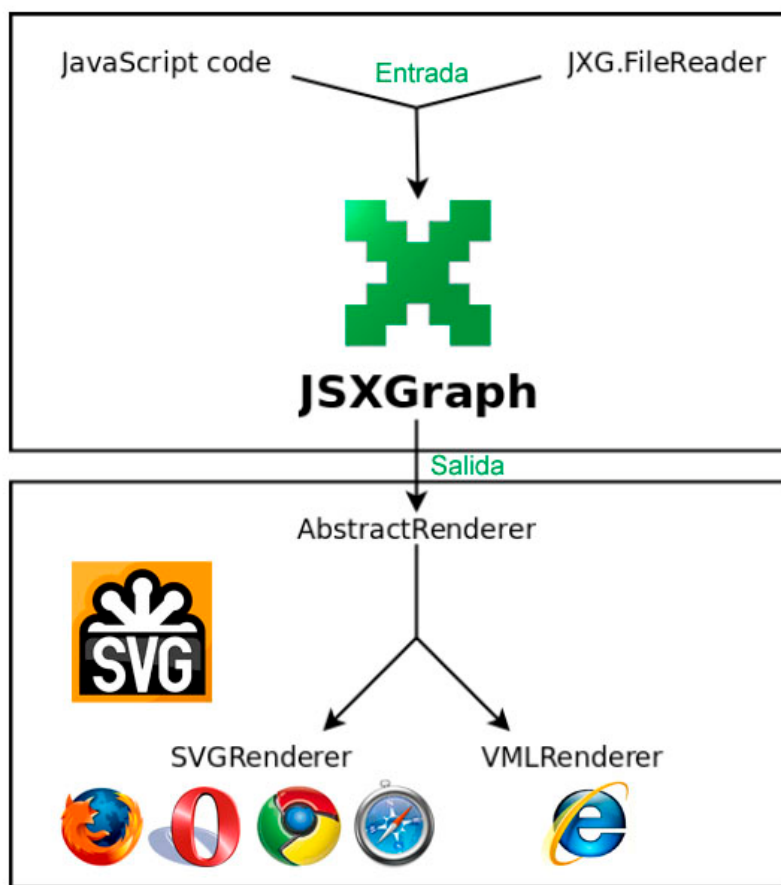


Figura 5.2: Renderización SVG o VML dependiendo del explorador [22].

Para Internet Explorer, al no soportar SVG, JSXGraph renderiza los gráficos con Vector Markup Language (VML), otro lenguaje XML de programación abierto, destinado a la creación de gráficos vectoriales. Sabiendo todo esto encontramos en este tipo de tecnologías una potencia realmente superior a la hora de realizar gráficos web. Algunas de las principales ventajas son las siguientes:

- Velocidad de lectura. La velocidad de lectura y representación de un gráfico en JavaScript + SVG es varias veces superior a la de una aplicación Java.
- Representación gráfica en dispositivos móviles. Gran cantidad de este tipo de dispositivos no utilizan Java/Flash.
- Plugins. JSXgraph no necesita instalación de plugins.

5.2.2.2. ¿Como representar gráficos en un navegador sin usar Java o Flash?

Una de las principales ventajas de JavaScript es su gran compatibilidad con la mayoría de los navegadores del mercado. Las tecnologías usadas para este tipo de representaciones por la librería JSXGraph son [23]:

- SVG (Firefox 2.0+, Chrome, Safari 3+, Opera, iPhone 2.1+)
- VML (IE 5+)

5.2.3. JSXGraph

Recopilando la información obtenida hasta ahora de la librería JSXGraph, tenemos que [23]:

- Es un proyecto de código abierto disponible en sourceforge.net.
- Está desarrollado en la Universidad de Bayreuth (Alemania).
- Implementado completamente en JavaScript.
- No requiere plugins.
- Soporta todos los principales navegadores (incluyendo IE).

Las principales características técnicas de esta librería son:

- JSXGraph necesita un ancho de banda de unos 80 kbytes.
- El tiempo de inicialización es cero.
- Integración perfecta en la página web con acceso dinámico a todos los elementos HTML.
- No requiere el uso del servidor web.
- La comunicación con el servidor es muy simple a través de AJAX.
- Una de las principales características funcionales del lenguaje de programación JavaScript es mejorar la interactividad con el usuario.

Para finalizar, otras características a destacar serían:

- Muchos de los elementos que utiliza JSXGraph están predefinidos, pero hay gran variedad de elementos ajustables.
- La interactividad es uno de sus grandes puntos fuertes.
- Puede realizar animaciones.
- Tiene su propio sistema de coordenadas.
- Funciona tanto con jQuery como con Prototype, y puede ser compatible con otras librerías.
- JSXGraph se puede utilizar en todos los dispositivos que ejecutan al menos uno de los navegadores compatibles, es decir [20]:
 - PC (Windows, Linux, Mac). Utilizando SVG, y VML para Internet Explorer.
 - Touchpads, como el iPad, permitiendo realizar eventos de manipulación táctil para el arrastre e incluso algunas de las acciones multitáctiles.
 - Teléfonos móviles, porque no requiere de ningún plugin para su funcionamiento (Ver figura 5.3).
 - Dispositivos Android, con un CanvasRenderer (disponible desde la versión 0.82rc8, aún en desarrollo).



Figura 5.3: JSXGraph en dispositivo móvil.

5.3 Hojas de estilo CSS.

Las Hojas de Estilo en Cascada (Cascading Style Sheets, CSS), son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla o cómo se va a imprimir. Esta forma de descripción de estilos nos ofrece el control total sobre el estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS nos permite controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca este elemento [24].

5.3.1. Estructura de la hoja de estilos.

Para la implementación de los estilos de la aplicación web se han utilizado dos estructuras (o plantillas) css. La primera es la más común en la aplicación, ya que se utiliza en todas las páginas excepto las relacionadas con el editor gráfico. Está compuesta por:

- La cabecera, que es una capa flotante que contiene el logo de la aplicación y los enlaces a las secciones principales.
- El menú, que contiene todos los enlaces de cada sección, así como una breve información acerca de la sesión iniciada por el terapeuta.
- El contenido, en el que se muestran todos los elementos relativos a la sección donde se navega.
- El pie de página.

La segunda plantilla es la que se utiliza para mostrar el editor de potenciales, la cual no tiene menú lateral y, además, la cabecera no dispone de las secciones principales. El contenido de estas clases es el siguiente:

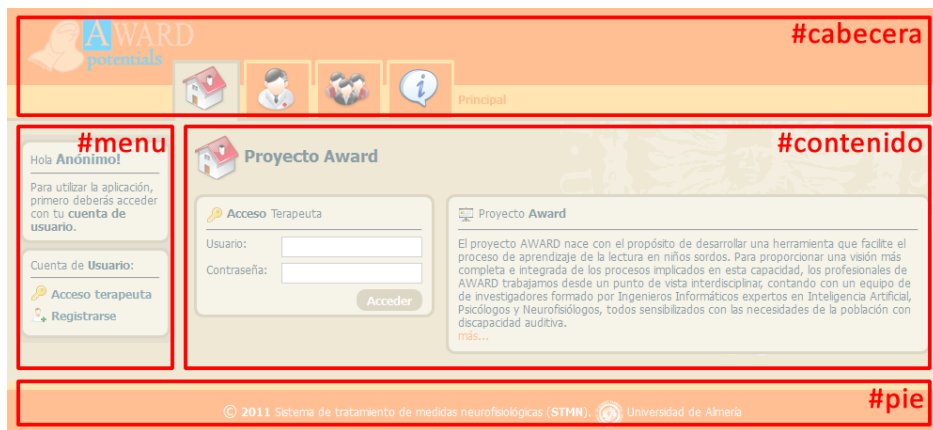


Figura 5.4: Primera plantilla CSS de la aplicación.

- La cabecera, donde se muestra información útil acerca del gráfico que se está visualizando y del sujeto al que corresponde, así como unos enlaces de retorno relacionados con la misma.
- El contenido, donde se muestra el editor gráfico junto con todos sus elementos de edición.
- El pie de página.

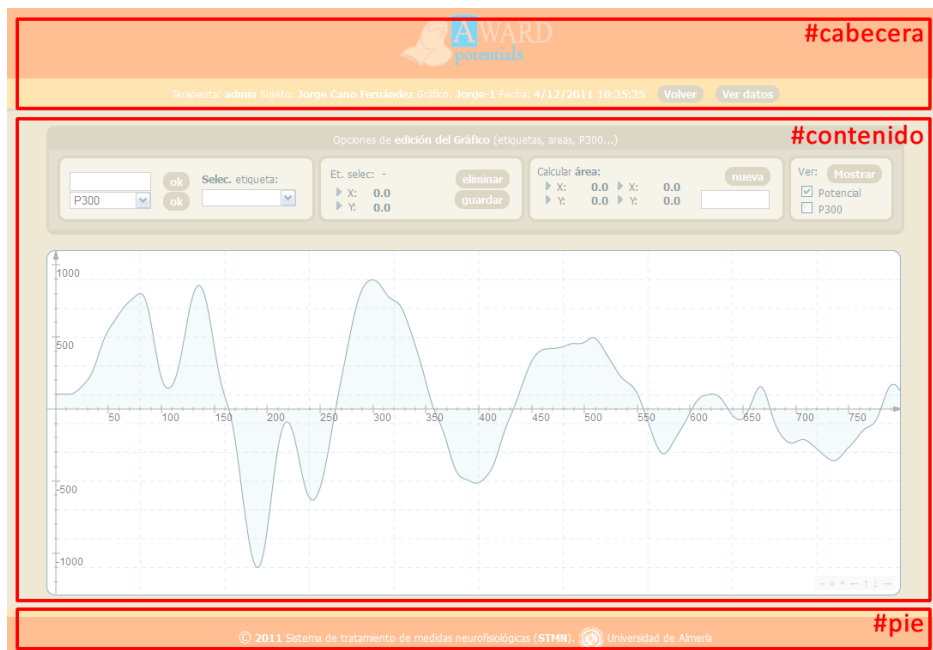


Figura 5.5: Segunda plantilla CSS de la aplicación.

5.3.2. Tipos de elementos CSS implementados.

Las hojas de estilo en cascada tienen una estructura muy simple pero muy flexible y potente. Gracias a esto hemos definido la apariencia de cada elemento o grupo de ellos

con total comodidad. Podemos definir distintas apariencias en función del medio por el que se mostrarán y cambiarlas posteriormente, si es necesario, de una forma muy simple y rápida [25].

Los tipos de elementos que hemos utilizado en nuestras hojas de estilos se pueden dividir en: clases, identificadores, pseudoclasas y pseudoelementos. A continuación pasamos a describir la forma de uso de cada uno de ellos.

5.3.2.1. Clases.

La clase es el nombre que se le asigna a una etiqueta HTML por medio del atributo CLASS para luego poder hacer referencia a ella. Gracias a las clases se pueden asignar propiedades a sólo una parte de los selectores de un mismo tipo. Por ejemplo, si hemos definido una o más tablas por medio de la etiqueta `<table class="listado">`, podemos asignarle las propiedades que queramos con la siguiente regla:

```

1 table.listado {
2     border: 4px solid #baad8b; // Grosor y color del borde
3     text-align: center; // Alineación del texto
4     margin-top: 5px; // Margen superior
5     -moz-border-radius: 10px; // Redondeo de las esquinas
6     -webkit-border-radius: 10px; // Redondeo de las esquinas
7 }

```

Esta regla afectará únicamente a los párrafos que tengan en su atributo CLASS el valor "listado", sin modificar los demás. La clase se especifica escribiéndola tras el selector seguido por un punto (sin dejar espacios). Si omitimos el selector y escribimos sólo un punto y el nombre de la clase, la regla se asigna a todos los selectores que tengan ese nombre de clase (ya sean párrafos, cabeceras, enlaces, etc).

5.3.2.2. Identificadores.

Los identificadores, que se asignan a una etiqueta HTML con el atributo ID, funcionan igual que las clases, pero con la diferencia de que se pueden emplear sólo una vez en un documento HTML. Es decir, si se tiene un párrafo al que se ha identificado como `<p ID="nombre">`, ese es el único elemento, sea párrafo o no, que puede tener ese ID. Para asignar una regla a ese párrafo hemos usamos el carácter "almohadilla" (#) de la siguiente forma:

```

1 #logo {
2     float: left; // Alineación flotante a la izquierda
3     height: 70px; // Altura
4 }

```

Como es normal, se han utilizado muchos menos identificadores que clases.

5.3.2.3. Pseudoclasas.

Las pseudoclasas son un tipo especial de clase que no definimos nosotros, si no que ya están definidas por el estandar CSS. Actualmente sólo existen cuatro pseudoclasas, y únicamente afectan al elemento <a>(a:link, a:active, a:visited, a:hover), que se corresponden, respectivamente, con los link que aún no han sido visitados (a:link), los que están activos mientras el navegador busca la dirección a la que hace referencia (a:active), los que ya han sido visitados (a:visited) y los que tienen en un instante dado el cursor encima (a:hover). Funcionan igual que las clases, pero usando dos puntos para separarlos del selector.

```
1 a:hover {
2     color:#FF7F26;
3     text-decoration: underline;
4 }
5 a:active {
6     color:#FF7F26;
7     text-decoration: none;
8 }
```

5.3.2.4. Pseudoelementos.

Los pseudoelementos son un tipo especial de clase que describe elementos particulares de la tipografía. Sólo existen dos en CSS1, y los navegadores actuales los ignoran casi por completo. Se pueden aplicar a cualquier selector de texto y se escriben igual que las pseudoclasas, es decir, usando dos puntos para separarlos del selector. En concreto:

```
1 SELECTOR:first-line
2 SELECTOR:first-letter
```

El pseudoelemento “first-line” se refiere a la primera línea del selector y “first-letter” a la primera letra. Por ejemplo, para asignar un color rojo a la primera letra de las cabeceras de primer nivel usaríamos:

```
1 H1:first-letter {
2     color: #ff0000;
3 }
```

5.3.3. Plantillas CSS utilizadas.

Para la implementación de este proyecto se han utilizado 5 plantillas CSS, de las cuales, tres han sido plantillas descargadas pertenecientes a distintos componentes JavaScript externos.

Las plantillas CSS implementadas por nosotros han sido:

- `estilo.css`, para los estilos de la estructura general de la aplicación.
- `estilo_grafico.css`, para los estilos de los gráficos de potenciales evocados.

Las plantillas pertenecientes a los diferentes scripts utilizados han sido:

- `ingareaselect-default.css`, para los estilos del selector de áreas utilizado en el perfil de los sujetos para recortar las imágenes de los mismos.
- `jsDatePick_ltr.min.css`, para los estilos del calendario.
- `jsxgraph.css`, para los estilos de los gráficos del editor de potenciales evocados.

5.3.4. Especificaciones de CSS.

CSS tiene diferentes niveles y perfiles. En líneas generales, los exploradores de escritorio implementan el nivel 1, 2 ó 3; otros programas implementan el perfil apropiado para su plataforma: el teléfono móvil, PDA, televisión, impresora o el sintetizador de habla entre otros [26].

- El nivel 1 de CSS (1996, 1999) incluye propiedades de fuentes, márgenes, colores, etc., que casi todas requieren CSS.
- El nivel 2 de CSS, revisión 1 (“CSS 2.1”) contiene todo lo del nivel 1 de CSS y agrega elementos completamente colocados, numeración automática, saltos de página, texto que se puede leer de derecha a izquierda y demás.
- El nivel 3 de CSS está en desarrollo. Incluye todo el nivel 2 y lo extiende con nuevos selectores, bordes y fondos agradables, texto vertical, interacción con el usuario, habla y mucho más.
- Perfil móvil de CSS 1.0. Está destinado a dispositivos como teléfonos móviles y PDA.
- Perfil de impresión de CSS (aún en fase borrador). Está dirigido a las impresoras de bajo costo.
- Perfil de televisión de CSS 1.0. Está destinado a los exploradores que se ejecutan en televisores.

En nuestro caso, hemos utilizado el nivel 3 de CSS, aunque algunas de las características visuales de este estilo no es apreciable desde según que explorador.

Otras especificaciones de las CSS incluyen:

- SVG, que contiene las propiedades de las CSS para dar estilo a gráficos vectoriales.
- Características de accesibilidad de CSS. Acerca de cómo las CSS pueden hacer los documentos más accesibles.
- Traducciones del nivel 1 de CSS y traducciones del nivel 2 de CSS.

5.4 Funciones del editor gráfico en JavaScript.

Como ya hemos comentado anteriormente, el editor gráfico de la aplicación está implementado íntegramente en JavaScript y usa la librería JSXGraph. Las funciones que se han implementado están repartidas en tres archivos dependiendo del tipo de gráfico o de la funcionalidad que se necesite en cada momento por la aplicación. Estos archivos son los siguientes:

- `funciones_grafico.php` para los gráficos que se muestran en el editor gráfico de forma individual.
- `funciones_grafico_multiple.php` para los distintos gráficos que se muestran a la vez en el editor gráfico.
- `funciones_grafico_mini.php` para los gráficos que se muestran en el menú de los sujetos.

Como se puede observar, aunque las funciones están implementadas en JavaScript, los archivos tienen extensión “.php”. Esto es debido a que la generación de ciertas partes del código JavaScript se ha hecho de forma dinámica según los datos obtenidos por los gráficos en PHP.

Por ejemplo, para la obtención del array con los datos de un gráfico en JavaScript, la forma de implementación en PHP ha sido la siguiente:

```

1 | <?php
2 |     $grafico = new clase_grafico($idPat);
3 |
4 |     echo "var array_datos = [";
5 |     echo $grafico->getDatos();
6 |     echo "];";
7 | ?>

```

Con lo que obtenemos la siguiente instrucción JavaScript:

```

1 | var array_datos = [1,2,3,4,5,...];

```

Este método de implementación se ha utilizado en distintas partes de la aplicación, ya que nos permite utilizar toda la información almacenada en la base de datos y recuperarla a través de funciones PHP para la síntesis de funciones escritas en JavaScript.

A continuación, se muestran algunas de las funciones más interesantes implementadas para el editor gráfico.

5.4.1. Función insertar etiqueta.

Esta función nos permite insertar una etiqueta asociada al gráfico escogiendo previamente el nombre de la misma mediante dos formas distintas. La primera forma consiste en escoger un nombre de etiqueta habitual para potenciales evocados a partir de una

lista desplegable (P300, N1, N2, ...). La segunda forma consiste en permitir al terapeuta asignar un nombre de etiqueta mediante un campo input de texto.

El código de la función asociada sería el que se muestra a continuación:

```

1 function insertarEtiqueta(nombreEtiqueta) {
2   brd.suspendUpdate();
3   // Crea el punto editable con el nombre de la etiquea.
4   var glider_etiqueta = brd.createElement('glider', [3, y[3], grafico_actual], {←
      visible:true, strokeWidth: 2, strokeColor: '#163e4d', highlightStrokeColor: '←
      #163e4d', fillColor: 'orange', highlightFillColor: 'yellow', style: 6, name: ←
      nombreEtiqueta, fixed:false});
5   // Inserta la nueva etiqueta en array_glidens_etiquetas.
6   array_glidens_etiquetas[nombreEtiqueta] = glider_etiqueta;
7   brd.unsuspendUpdate();
8
9   // Mete el nombre de la Etiqueta en el Select utilizando DOM.
10  var select = document.getElementById("selectEtiquetas");
11  var opcion = document.createElement("option");
12  opcion.appendChild(document.createTextNode(nombreEtiqueta));
13  select.appendChild(opcion);
14  // Pone como seleccionada la nueva etiqueta.
15  opcion.selected = true;
16
17  // Selecciona la etiqueta insertada. Esta función muestra las propiedades de la ←
18  nueva etiqueta en el menú de edición.
19  seleccionarEtiqueta(nombreEtiqueta);
20 }

```

Un ejemplo del resultado al usar esta función sería el siguiente.

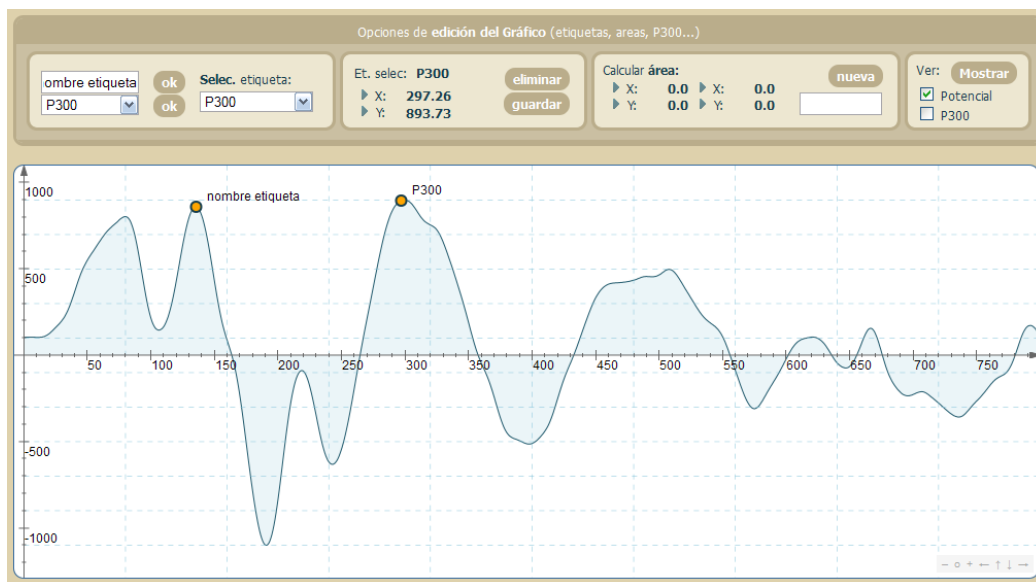


Figura 5.6: Resultado de la función insertarEtiqueta().

5.4.2. Función insertar area.

La función insertarArea() nos permite insertar un polígono con la forma del área comprendida entre dos puntos editables por el terapeuta. El área generada muestra un

valor de su superficie, el cual se va recalculando automáticamente mientras el terapeuta mueve alguno de los puntos en sus extremos.

Esta función es muy útil para los terapeutas, ya que permite conocer el valor de la superficie de un componente o pico, ya sea positivo o negativo, el cual indica un valor objetivo asociado con los potenciales.

A continuación se puede ver el código de la función `insertarArea()` el cual, básicamente, está dividido en dos partes: (1) la creación de los puntos (*gliders*) y el área que los delimitan; y (2) la parte donde se describe el método de actualización del polígono junto con el valor de la superficie cuando el terapeuta mueve alguno de los *gliders* de los extremos.

```

1 function insertarArea() {
2     // Comprueba que no hay ningun área dibujada en el panel de edición.
3     if (!setArea) {
4         setArea = true;
5         brd.suspendUpdate();
6
7         // Crea los dos puntos de los extremos (gliders) del área Xmin y Xmax.
8         var g1 = brd.createElement('glider', [x_actual[340], y_actual[340], ←
          grafico_actual], {visible:true, strokeWidth: 2, strokeColor: '#ffffff', ←
            highlightStrokeColor: '#3796b5', fillColor: '#163e4d', highlightFillColor: ←
              '#163e4d', style: 5, name: 'X_min', fixed:false});
9         var g2 = brd.createElement('glider', [x_actual[420], y_actual[420], ←
          grafico_actual], {visible:true, strokeWidth: 2, strokeColor: '#ffffff', ←
            highlightStrokeColor: '#3796b5', fillColor: '#163e4d', highlightFillColor: ←
              '#163e4d', style: 5, name: 'X_max', fixed:false});
10
11        // Crea un area vacía.
12        var area = brd.createElement('curve', [[0],[0]], {strokeColor:'#163e4d', dash←
          :0, fillColor:'#3796b5', fillOpacity:0.3, highlightFillColor:'#3796b5', ←
            highlightStrokeColor: '#163e4d', highlightFillOpacity:0.3});
13
14        // Función que modifica el array de puntos que forman el área según el ←
          movimiento de los gliders.
15        area.updatedataArray = function() {
16
17            // Crea las variables up y down con los valores de los extremos según la ←
          escala del gráfico.
18            var up_array = g2.X();
19            var down_array = g1.X();
20            var up = parseInt(g2.X() / tam_iteracion);
21            var down = parseInt(g1.X() / tam_iteracion);
22
23            // Actualiza los valores de los extremos x e y, y los muestra en pantalla ←
          usando DOM.
24            document.getElementById("punto_min_x").innerHTML = Math.floor(up_array * ←
          100)/100;
25            document.getElementById("punto_min_y").innerHTML = y_actual[up];
26            document.getElementById("punto_max_x").innerHTML = Math.floor(down_array * ←
          100)/100;
27            document.getElementById("punto_max_y").innerHTML = y_actual[down];
28
29            // Lee los valores entre los gliders y rellena los arrays dataX y dataY del←
          área.
30            if(down <= up) {
31                this.dataX = x_actual.slice(down, up + 1);
32                this.dataY = y_actual.slice(down, up + 1);
33
34                // Calcula el valor del área y lo muestra usando DOM.
35                calcularMostrarArea(array_datos_actual, down, up);
36
37                // Rellena los puntos extremos que conforman el area.
38                this.dataX.push(up_array);
39                this.dataY.push(0);

```

```

40     this.dataX.push(down_array);
41     this.dataY.push(0);
42     this.dataX.push(down_array);
43     this.dataY.push(y_actual[down]);
44   } else {
45     this.dataX = x.slice(up, down + 1);
46     this.dataY = y.slice(up, down + 1);
47
48     // Calcula el valor del área y lo muestra usando DOM.
49     calcularMostrarArea(array_datos_actual, up, down);
50
51     // Rellena los puntos extremos que conforman el area.
52     this.dataX.push(down_array);
53     this.dataY.push(0);
54     this.dataX.push(up_array);
55     this.dataY.push(0);
56     this.dataX.push(up_array);
57     this.dataY.push(y_actual[up]);
58   }
59
60   };
61   brd.unsuspendUpdate();
62 }
63
64 }

```

Visualmente, el resultado de la función sería el que se muestra en la siguiente figura, donde podemos ver el área delimitada entre los puntos X_{min} y X_{max} , y en la parte superior derecha los datos de posicionamiento X e Y de los dos puntos, además del resultado del cálculo de la superficie.

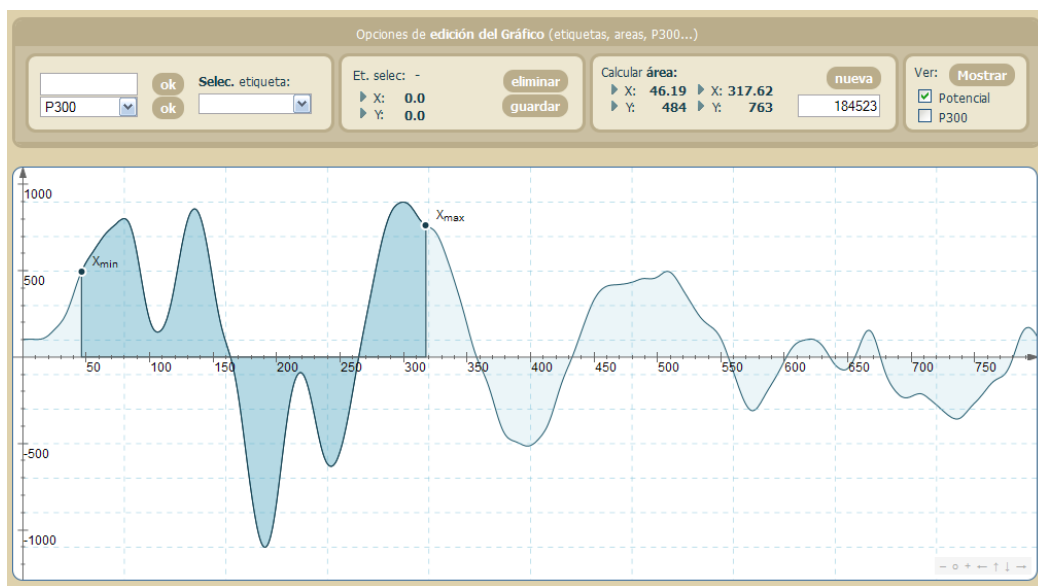


Figura 5.7: Resultado de la función `insertarArea()`.

5.4.3. Función insertar distancia.

Esta función, como su propio nombre indica, nos permite conocer la distancia entre dos puntos de distintos gráficos (o del mismo si así se desea). Esta funcionalidad es realmente útil si se está realizando un estudio evolutivo en un mismo sujeto y se quiere comprobar si

se está produciendo una mejora. Por ejemplo, para un potencial evocado visual, se podría calcular la distancia entre los componentes N1 de distintos test y poder ver si se está reduciendo para comprobar si la velocidad de reacción del sujeto está experimentando una mejora.

La estructura del código de la función es similar a la de la función `insertarArea()`, ya que se divide en dos partes. La primera muestra los puntos de los extremos editables por el usuario y la línea (distancia) que los une. La segunda parte del código se encarga de ir recalculando la distancia (así como dibujarla gráficamente) en el caso de que el terapeuta realice un cambio de posición en alguno de los *gliders*.

```

1 function insertarDistancia() {
2     brd.suspendUpdate();
3
4     // Obtiene los dos gráficos para los que se va a insertar la distancia.
5     var iDesde = document.getElementById("selectMedidorDesde").selectedIndex;
6     var iHasta = document.getElementById("selectMedidorHasta").selectedIndex;
7
8     // Crea los dos puntos extremos (gliders) de la línea distancia.
9     var g1 = brd.createElement('glider', [array_x[iDesde][340], array_y[iDesde][340], ←
10        array_graficos[iDesde]], {visible:true, strokeWidth: 2, strokeColor: '#3796b5', ←
11        fillColor: '#163e4d', style: 2, name: 'X_m_i_n', fixed:false});
12     var g2 = brd.createElement('glider', [array_x[iHasta][420], array_y[iHasta][420], ←
13        array_graficos[iHasta]], {visible:true, strokeWidth: 2, strokeColor: '#3796b5', ←
14        fillColor: '#163e4d', style: 2, name: 'X_m_a_x', fixed:false});
15
16     // Crea la distancia.
17     var linea = brd.createElement('line', [g1,g2], {strokeColor:'#163e4d', dash:1, ←
18        fillColor:'#3796b5', fillOpacity:0.3, highlightFillColor:'#3796b5', ←
19        highlightStrokeColor: '#163e4d', highlightFillOpacity:0.3, straightFirst: false ←
20        , straightLast: false, draft:true});
21
22     // Función que actualiza los valores intermedios de la línea según los puntos de ←
23     los extremos.
24     linea.update = function() {
25
26         // Inserta con DOM los valores X e Y de los dos puntos extremos.
27         document.getElementById("punto_min_x").innerHTML = Math.floor(g1.X() * 100) ←
28             /100;
29         document.getElementById("punto_min_y").innerHTML = Math.floor(g1.Y() * 100) ←
30             /100;
31         document.getElementById("punto_max_x").innerHTML = Math.floor(g2.X() * 100) ←
32             /100;
33         document.getElementById("punto_max_y").innerHTML = Math.floor(g2.Y() * 100) ←
34             /100;
35
36         // Calcula la distancia entre los dos puntos.
37         var dist = 0;
38         if (g1.X() > g2.X()) dist = g1.X() - g2.X();
39         if (g1.X() < g2.X()) dist = g2.X() - g1.X();
40         dist = Math.floor(dist * 100)/100;
41
42         // Muestra la distancia usando DOM.
43         document.getElementById("distancia").setAttribute("value", dist);
44     }
45     brd.unsuspendUpdate();
46 }

```

Como se observa en la siguiente figura, la distancia entre los dos puntos seleccionados se muestra en los dos gráficos del editor. En la parte superior derecha vemos toda la información relativa.

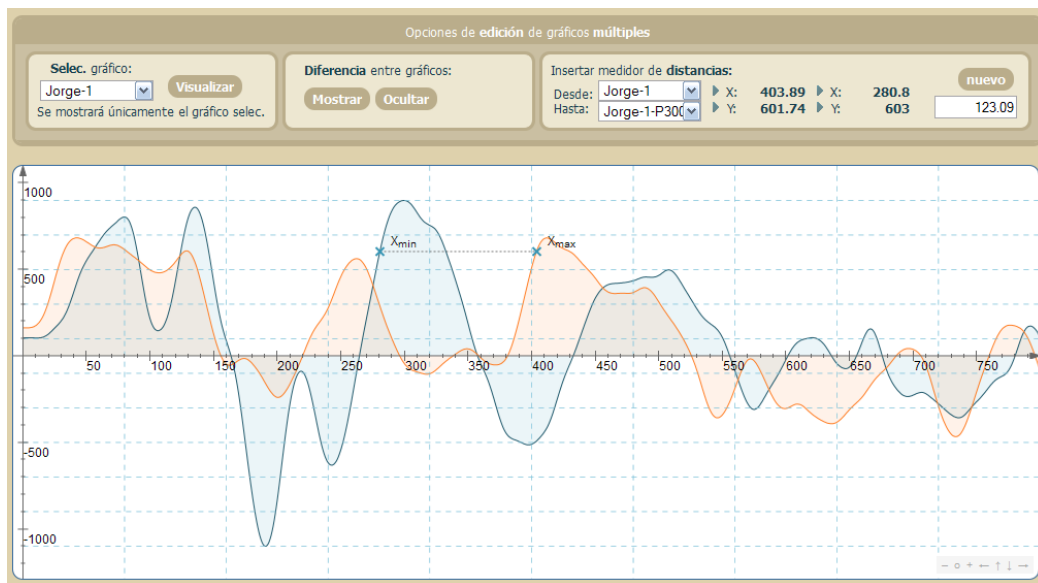


Figura 5.8: Resultado de la función insertarDistancia().

5.4.4. Función mostrar diferencia.

Esta función es de las más interesantes a la hora de realizar un estudio de obtención de potenciales evocados. Se encarga de generar una nueva gráfica con la diferencia de distancia entre cada punto de dos o más test de potenciales cognitivos o *P300*.

Su funcionalidad es clave para dos casos:

- Comparación de varios test del mismo tipo. Para este caso, si por ejemplo utilizamos la función `mostrarDiferencia()` para dos test auditivos realizados el mismo día, al mismo paciente, deberíamos visualizar un resultado que fuera prácticamente nulo (todos los puntos del gráfico deberían tener valor 0), ya que el sujeto en un periodo tan corto de tiempo no debería haber mejorado/empeorado ninguna de sus características neurofisiológicas.
- Comparación de potencial con su *P300*. En este caso, lo ideal sería que al utilizar la función `mostrarDiferencia()`, el resultado fuese todo 0 excepto el componente *P300* (a los 300ms y alrededores) donde debería haber un pico claramente distinguido.

La función básicamente recoge los datos de todos los gráficos para los que se quiere obtener la diferencia y va calculando punto a punto la misma. La forma de obtener los datos de todos los gráficos es mediante PHP como ya se explicó al principio de esta sección.

```

1 function mostrarDiferencia() {
2
3     // Crea un array que contendrá a su vez tantos arrays de datos como gráficos haya ←
4     // para calcular la diferencia.
5     var array_datos_todos = new Array();
6
7     // Mediante impresión de código JavaScript usando PHP, imprimimos los arrays de ←
8     // datos de todos los gráficos.

```

```

7 <?php
8 $graficos = array();
9 for ($i = 0; $i < count($idPat); $i++) {
10     $graficos[$i] = new clase_grafico($idPat[$i]);
11
12     // Creamos los arrays asignandoles al final del nombre de la variable un ←
13     // identificador numérico.
14     echo "var array_datos" . $i . " = [";
15     echo $graficos[$i]->getDatos();
16     echo "];";
17
18     // Cada array de datos creado lo introducimos en el array general.
19     echo "array_datos_todos[" . $i . " ] = array_datos" . $i . " . ";";
20 }
21 ?>
22 // Inicializamos el array que contendrá la diferencia entre gráficos con los datos ←
23 // del primero de ellos.
24 var array_diferencia = array_datos0;
25
26 // Por cada gráfico que haya y cada punto del mismo vamos calculando la diferencia ←
27 // y generando el array_diferencia.
28 for (i = 1; i < <?php echo count($idPat); ?>; i++) {
29     for (j = 0; j < 1000; j++) {
30         if (array_diferencia[j] > array_datos_todos[i][j]) {
31             array_diferencia[j] = array_datos_todos[i][j];
32         } else {
33             array_diferencia[j] = array_datos_todos[i][j] - array_diferencia[j];
34         }
35     }
36 }
37 // Imprimimos el gráfico diferencia resultante.
38 imprimirGraficoDiferencia(array_diferencia);
39 }

```

En la siguiente figura podemos observar el resultado de la función que calcula la diferencia entre gráficos. Para el ejemplo de la imagen se usó un test de potencial visual con su P300, por lo que lo ideal habría sido que el único pico de gran altura resultante fuese el de los 300ms.

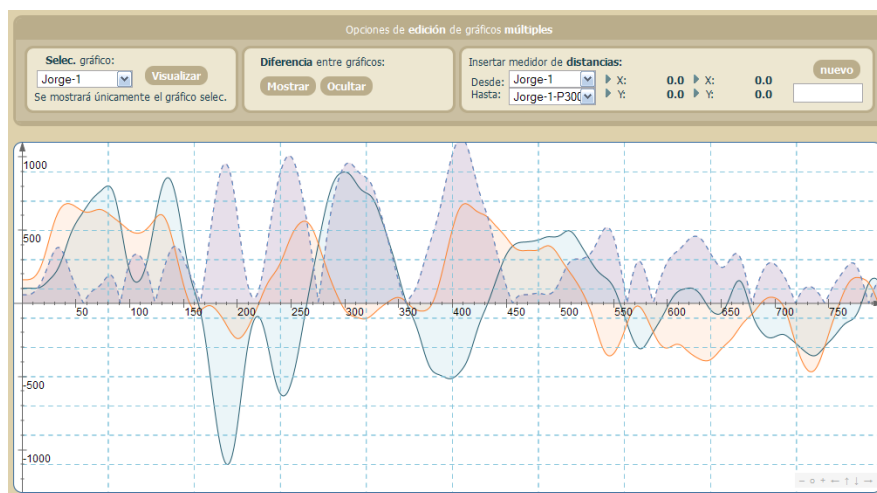


Figura 5.9: Resultado de la función mostrarDiferencia().

5.5 Aplicación web rica usando AJAX.

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como “JavaScript asíncrono + XML”. Pero AJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes [27].

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

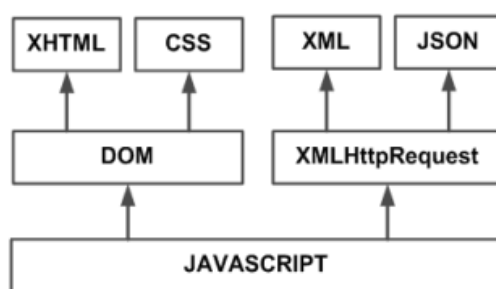


Figura 5.10: Tecnologías agrupadas bajo el concepto AJAX.

Desarrollar aplicaciones AJAX profesionales requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (hacer clic en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador cliente. En la figura 5.11, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX.

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página ya que el intercambio de información con el

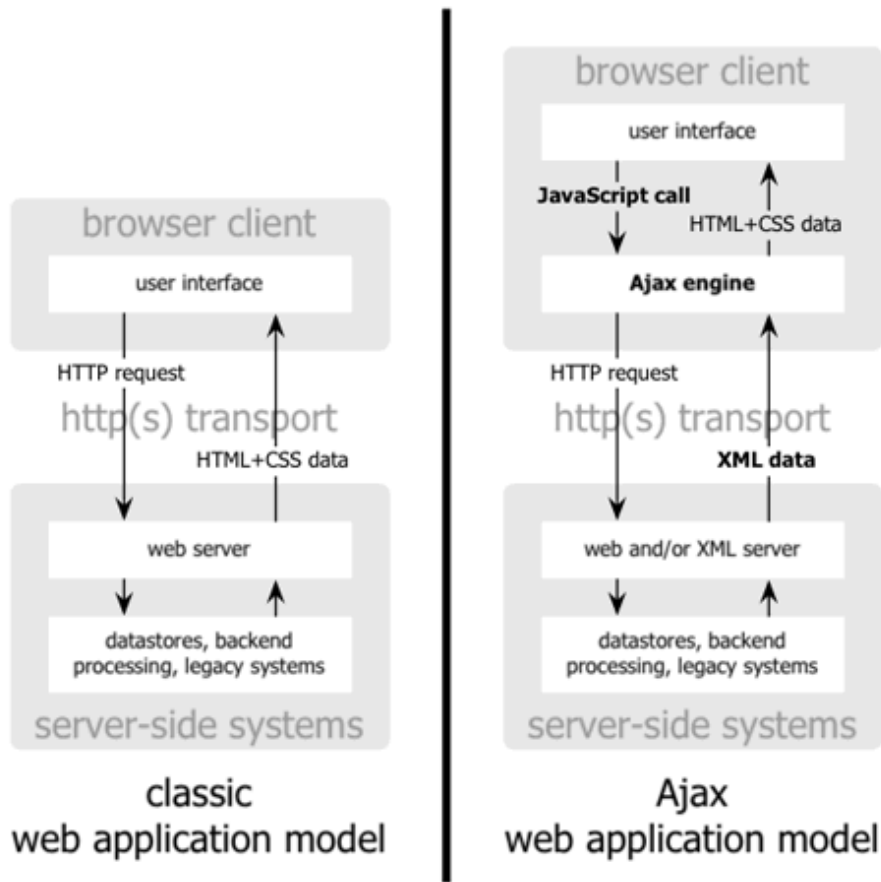


Figura 5.11: Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX.

servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

El siguiente esquema muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

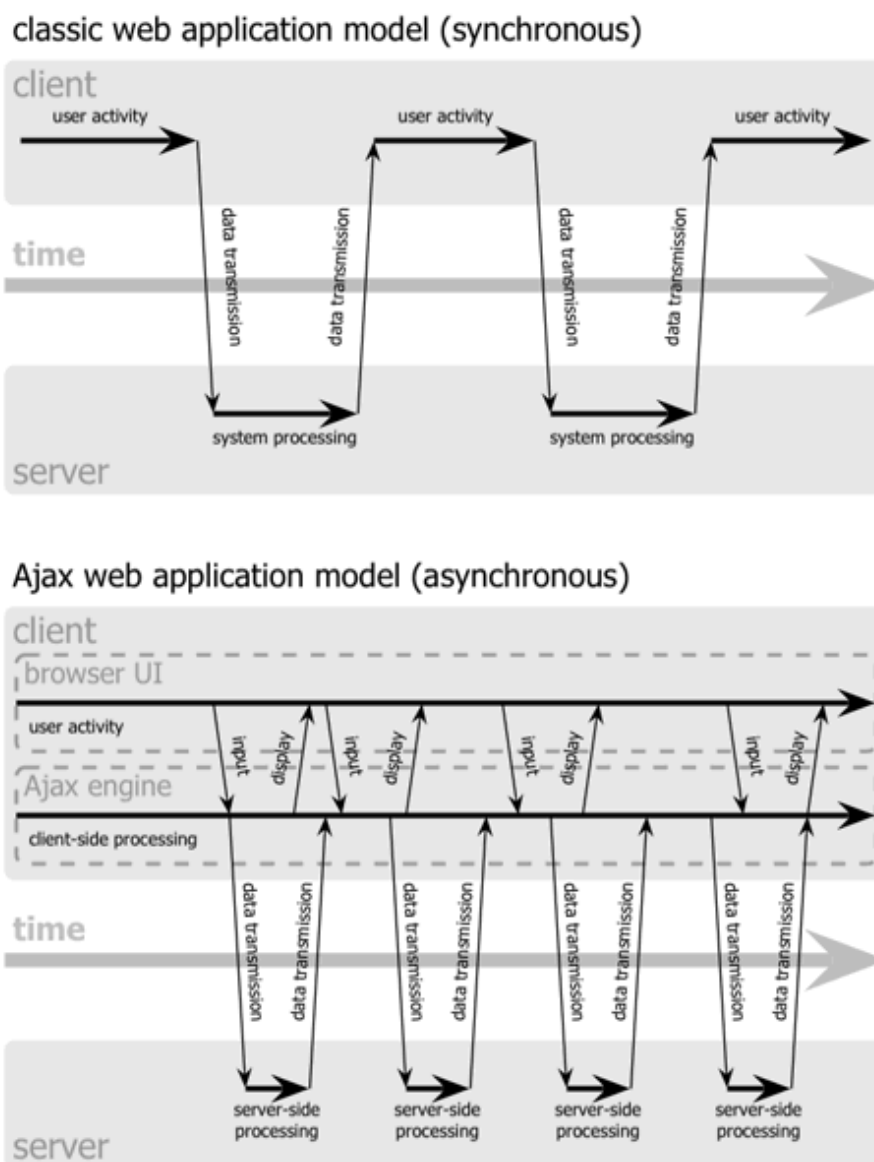


Figura 5.12: Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX.

5.5.1. El objeto *XMLHttpRequest*.

XMLHttpRequest es lo que la mayoría de los navegadores denominan el objeto petición u objeto *request*, el cual puede ser enviado al servidor para procesar posteriormente su respuesta sin necesidad de volver a cargar la página de forma completa. Así pues, en una aplicación AJAX, las peticiones al servidor se realizan mediante el objeto *XMLHttpRequest*. [28]

Las propiedades principales del objeto *XMLHttpRequest* son:

- **onreadystatechange.** Almacena el nombre de la función que se ejecutará cuando el objeto *XMLHttpRequest* cambie de estado.

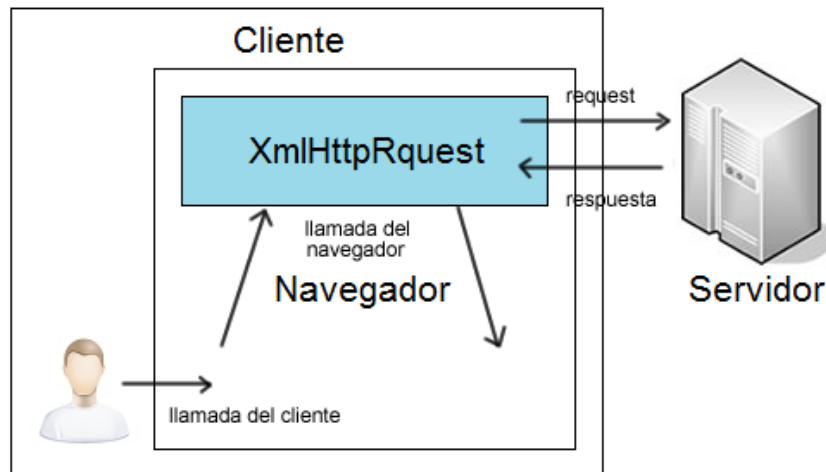


Figura 5.13: Funcionamiento del objeto *XMLHttpRequest*.

- **readyState**. Almacena el estado de la solicitud realizada al servidor, pudiendo ser:
 - 0. No inicializado (el método `open` no ha sido llamado).
 - 1. Cargando (se llamó al método `open`).
 - 2. Cargado (se llamó al método `send` y ya tenemos la cabecera de la petición HTTP junto el *status*).
 - 3. Interactivo (la propiedad `responseText` tiene datos parciales).
 - 4. Completado (la propiedad `responseText` tiene todos los datos solicitados).
- **responseText**. Almacena el string devuelto por el servidor después de haber realizado una petición.
- **responseXML**. Similar a la anterior (`responseText`), con la diferencia que el string devuelto por el servidor se encuentra en formato XML.

Los métodos principales del objeto *XMLHttpRequest* son:

- **open**. Abre una solicitud HTTP al servidor.
- **send**. Envía la solicitud al servidor.

Para nuestra aplicación se ha creado un archivo JavaScript denominado “`utils.js`”, el cual contiene la función que crea el objeto *XMLHttpRequest* dependiendo del navegador desde donde se realice la llamada. Hay que tener en cuenta que en las versiones de Internet Explorer anteriores a la 7, este objeto no se encuentra implementado de forma nativa, por lo que hay que crearlo como objeto *ActiveX*.

```

1 function crearRequest(){
2     try {
3         // Creamos el objeto XMLHttpRequest si el navegador lo implementa de forma nativa.
    
```

```

4     request = new XMLHttpRequest();
5 } catch (intentarMS) {
6     try {
7         // Si no ha podido crearlo, intenta hacerlo como objeto ActiveX.
8         request = new ActiveXObject("Msxml2.XMLHTTP");
9     } catch (otroMS) {
10        try {
11            // Si aún no se ha creado el objeto, lo intenta de nuevo (posiblemente ←
12                IE6 o menor).
13            request = new ActiveXObject("Microsoft.XMLHTTP");
14        } catch (error) {
15            // No se puede crear y devuelve null.
16            request = null;
17        }
18    }
19 }
20 return request;
21 }

```

5.5.2. Uso de la tecnología AJAX en *AWARD*^{Potentials}.

Para nuestra aplicación, se han implementado diversas utilidades basadas en la tecnología AJAX, aunque todas ellas siguen la misma estructura. A grandes rasgos, se trata de una función que crea el objeto *request* y realiza una petición, y otra que tramita la respuesta. A continuación se detalla el proceso con más exactitud:

1. Se crea el objeto *XMLHttpRequest*.
2. Se crea la función de petición indicando que debe ser asíncrona e indicamos el archivo *.php* que debe devolver la información solicitada. Después se envía el objeto *request* al servidor.
3. Se crea la función de respuesta, que comprueba el estado del objeto *request* y ejecuta las acciones pertinentes.

Como ejemplo, mostraremos el código de las funciones creadas para el script *validacion.js*, el cual se encarga de comprobar de forma asíncrona si el *id* para el sujeto introducido por el terapeuta es único o, por el contrario, en la base de datos ya existe un identificador igual (ver figura 5.14).

La función de petición es *comprobarCodigo()* y funciona de la siguiente forma:

```

1 function comprobarCodigo(){
2     // Crea el objeto XMLHttpRequest.
3     request = crearRequest();
4     // Si el objeto es null muestra mensaje de error.
5     if (request == null) {
6         alert("Error al crear el objeto request");
7     } else {
8         // Define la url del script en PHP que devuelve la información necesaria.
9         var codigo = document.getElementById("codigo").value;
10        var url = "./ajax/comprobarCodigo.php?codigo=" + codigo;
11        // Se abre el objeto request indicando el método de envío (GET), la url que ←
12            realiza la acción y un booleano que indica que la petición es asíncrona (←
13            true).
14        request.open("GET", url, true);

```

```

13     // Definimos la función de respuesta para cuando el servidor envíe su respuesta←
14     request.onreadystatechange = mostrarEstadoCodigo;
15     // Enviamos la petición.
16     request.send(null);
17 }
18 }

```

La función de respuesta debe comprobar únicamente si la propiedad del objeto *request*, *readyState* es igual a 4 (el estado actual del objeto es “completo”) y la propiedad *status* es igual a 200 (“OK” en forma codificada del estado HTTP devuelto por el servidor). De ser así, realiza las acciones pertinentes.

```

1 function mostrarEstadoCodigo(){
2     // Comprueba que el estado actual del objeto es "completo" (4).
3     if (request.readyState == 4) {
4         // Comprueba que el estado HTTP devuelto por el servidor es "OK" (200).
5         if (request.status == 200){
6             // De ser así muestra un icono verde y habilita el botón de registro.
7             if (request.responseText == "Disponible") {
8                 document.getElementById("estadoCodigo").innerHTML = "<img src='./img/ok←
9                 .png'> ";
10                document.getElementById("btnRegistrar").disabled = false;
11            // En caso contrario muestra un icono rojo, mueve el foco del ratón al ←
12            input del código y deshabilita el botón de registro.
13            } else {
14                document.getElementById("estadoCodigo").innerHTML = "<img src='./img/←
15                fail.png'> ";
16                document.getElementById("codigo").focus();
17                document.getElementById("btnRegistrar").disabled = true;
18            }
19        }
20    }
21 }
22 // Mientras el readyStrate no es 4, se muestra el icono "loading.png" (cargando).
23 else document.getElementById("estadoCodigo").innerHTML = "<img src='./img/loading.←
24     png'> ";
25 }

```

The image shows three instances of a registration form titled "Registro de nuevo sujeto:". Each form has input fields for "Nombre:", "Apellidos:", and "Fecha nacimiento:". The "Código:" field is pre-filled with a value and accompanied by a status icon.

- Top form:** "Código:" is 555 with a green checkmark icon, indicating a successful registration.
- Middle form:** "Código:" is 666 with a red minus sign icon, indicating a failed registration.
- Bottom form:** "Código:" is 444 with a loading icon (a circle with a dot), indicating the registration is in progress.

Figura 5.14: Ejemplo de script validación AJAX.

6

Evaluación de *AWARD*^{Potentials}.

En este capítulo trataremos tanto la prueba, actividad fundamental propuesta en RUP, como el proceso de implantación del sistema y la valoración que han realizado los usuarios de éste.

Según la metodología RUP, en el flujo de trabajo de la prueba del sistema se verifica el resultado de la implementación comprobando cada una de las construcciones, tanto las internas como las intermedias, así como las versiones finales del sistema a entregar a terceros. Sin embargo, tradicionalmente la prueba se ha asociado a la verificación y a la validación (V & V), que es el nombre que reciben los procesos de comprobación y análisis tendentes a asegurar que el software esté acorde con su especificación y cumpla con las necesidades de los clientes. Dichas constataciones se prolongan durante todo el proceso de ingeniería del software.

La verificación y la validación no son la misma cosa; la primera se refiere al conjunto de actividades que asegura que el software implementa correctamente una función específica, la actividad de prueba en RUP, mientras que la validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. Barry Boehm en 1981 las definió, respectivamente, como: *¿Estamos construyendo el producto correctamente?* (verificación) *¿Estamos construyendo el producto correcto?* (validación).

6.1 Validación.

En esta actividad se debe confirmar que los requisitos corresponden realmente a las necesidades de los terapeutas que van a utilizar la aplicación, que son los que realmente deben realizar esta actividad, aunque los desarrolladores de la aplicación pueden ayudar en el proceso mediante las explicaciones oportunas. Por tanto, la validación se ha reali-

zado dentro de la actividad fundamental de requisitos. El esfuerzo del desarrollador en esta tarea ha sido mínimo, puesto que tal y como se propone en [29], si los clientes y usuarios tienen conocimientos suficientes para comprender las notaciones utilizadas, se podrán utilizar en la validación de los modelos de análisis, es decir, en la validación de los casos de uso, los cuales son modificados por los clientes y reenviados a los desarrolladores. Este ha sido el caso del proyecto software *AWARD^{Potentials}*, ya que los requisitos han sido definidos por un equipo interdisciplinar formado por profesionales de la neuropsicología y neurofisiólogos, así como ingenieros informáticos con conocimientos sobre la teoría de potenciales evocados cognitivos así como del síndrome TDAH.

Las actividades de validación conllevan generalmente nuevos requisitos debido a que, a medida que el sistema se va perfilando, suelen ir apareciendo nuevas necesidades que hasta entonces estaban ocultas, sobre todo mediante la utilización de prototipos. Para *AWARD^{Potentials}* la validación se ha realizado en tres iteraciones. En la segunda y la tercera estaba disponible un prototipo que fue evaluado por el “cliente”. El resultado final fue la obtención de un catálogo de requisitos válido, casos de uso, y parte de la especificación de las clases de análisis, también validadas por los clientes, que permitió abordar el diseño de la aplicación final.

6.2 Verificación.

Una vez generado el software, debe ser probado para descubrir y corregir el máximo de errores posibles antes de su finalización. El objetivo consiste en diseñar los casos de prueba que tengan una alta probabilidad de encontrar errores. Estas son las llamadas pruebas de defectos, que se suelen acompañar de las pruebas estadísticas que se utilizan para verificar el desempeño y la fiabilidad del programa y comprobar cómo trabaja bajo condiciones operacionales, constatando los accesos de los usuarios y su frecuencia. Estas últimas se han puesto en práctica durante las tareas de aceptación por parte de los clientes/terapeutas.

Según [8], la prueba de defectos se realiza en dos grandes etapas:

- Las pruebas del componente (realizadas por el desarrollador). Comprenden las pruebas de funcionamiento de los componentes claramente identificables, que son funciones o grupos de métodos conjuntados en módulos o en objetos.
- Las pruebas de integración (realizadas por un equipo independiente). Durante estas pruebas, que para *AWARD^{Potentials}* también han sido realizadas por el desarrollador debido a la limitación de recursos, los componentes se integran para formar subsistemas o el sistema completo. En esta etapa las pruebas se enfocan en las interacciones entre componentes y en la funcionalidad y el desempeño del sistema como un todo.

En nuestra aplicación se han implementado tres niveles de pruebas:

1. Probar las operaciones individuales asociadas con los objetos. Se han aplicado tanto técnicas de caja blanca como de caja negra.

2. Probar clases de objetos individuales, aplicando pruebas de caja negra y teniendo en cuenta la estructura de módulos/clases.
3. Probar cluster de objetos. La integración estricta descendente o ascendente no es apropiada para crear grupos de objetos relacionados y se utiliza un enfoque basado en casos de uso. Cada caso de uso describe un escenario de utilización del sistema y un cluster de objetos que se prueban de forma integrada. Ejemplo: Alta o baja de sujetos de un proyecto.

6.3 Aceptación de los usuarios.

Una vez hemos obtenido la especificación del sistema que se va a construir, es necesario tener en cuenta que el sistema implementado debe ser aceptado por parte de los usuarios finales. Para ello, se han utilizado una serie de cuestionarios cuyos resultados aún no han sido recibidos.

6.3.1. Cuestionario de validación: explicación y funcionalidad.

Se ha desarrollado un cuestionario para valorar la aceptación del sistema por parte de los usuarios. Las siguientes preguntas y sus correspondientes respuestas ayudan a determinar si el sistema satisface las necesidades y requisitos de los usuarios y si la forma de interactuar con el sistema es adecuada.

Pregunta	Valoración (1-10)	Comentario
¿Considera que la distribución de contenidos es correcta?		
Al introducir nueva información, ¿sabe exactamente cómo y dónde hacerlo?		
Cuando se realiza una tarea con el sistema, ¿cree que puede acceder a toda la información necesaria?		
¿El sistema le ayuda con información adecuada? (mensajes de error, avisos...)		
¿Cree que el sistema le permite hacer todo lo que necesita?		
¿El sistema responde de forma apropiada a las acciones del usuario?		
¿El sistema justifica por qué ha solicitado una determinada información?		
¿Los mensajes de explicación son correctos?		
¿El sistema explica convenientemente las situaciones especiales?		

Tabla 6.1: Cuestionario de validación.

6.3.2. Cuestionario sobre oportunidad y previsiones futuras

Este cuestionario trata sobre la utilidad de este tipo de sistemas para terapeutas y profesionales de la psicología y neurofisiología. Además se les ofrece la oportunidad de realizar cualquier consideración sobre la oportunidad e implantación de sistemas como este en las instituciones médicas.

Pregunta	Comentario
¿Hacia quién cree que va dirigida esta aplicación?	
¿En el desarrollo del estudio de un sujeto con TDAH, ¿Para qué cree que le puede ayudar?	
¿Qué problema cree que resuelve mejor esta aplicación?	
¿En qué otros temas cree que se puede usar este tipo de herramientas basadas en internet?	
¿Conoce algún otro sistema o herramienta que tenga una finalidad parecida?	

Tabla 6.2: Cuestionario sobre oportunidad y previsiones futuras.

6.3.3. Críticas, mejoras, sugerencias y errores destacados.

En este cuestionario se le proporciona al usuario la oportunidad de especificar cuantas críticas o sugerencias estime oportunas, reflejando en cada momento las razones de las mismas.

Sugerencia	Objetivo	Justificación
Error Añadiría Mejoraría Falta Incorrecto Aclararía Cambiaría ...		
Error Añadiría Mejoraría Falta Incorrecto Aclararía Cambiaría ...		
...		

Tabla 6.3: Cuestionario de críticas, mejoras y sugerencias.

7

Conclusiones y trabajos futuros.

En este capítulo se enumerarán las conclusiones obtenidas tras el diseño e implementación de la herramienta *AWARD^{Potentials}*, así como los trabajos futuros propuestos para continuar en la línea de este proyecto.

7.1 Conclusiones.

A continuación se presentan de forma resumida las conclusiones obtenidas tras la finalización del proyecto:

- Se ha llevado a cabo el diseño e implementación de una aplicación web dinámica para la gestión y seguimiento evolutivo de pacientes a partir de medidas neurofisiológicas (potenciales evocados cognitivos).
- La aplicación web ofrece al terapeuta la posibilidad de gestionar información clínica de sujetos de manera intuitiva, rápida y sencilla. Permite la importación de señales neurofisiológicas adquiridas con un equipo de captura de potenciales evocados. El sistema permite editarlos para que el terapeuta pueda adquirir conocimiento y llevar un control evolutivo de pacientes a partir del conjunto de tests realizados así como poder llevar un control de las conclusiones diagnósticas obtenidas.
- La interfaz visual es simple, intuitiva y fácil de usar, permitiendo a terapeutas sin grandes conocimientos informáticos el uso de la aplicación sin dificultad alguna.
- El estar basada en una tecnología web dinámica, como PHP y JavaScript, ha permitido la creación de una aplicación de funcionalidad y usabilidad similares a una aplicación de escritorio, pero accesible desde un cliente ligero como es un navegador, eliminando necesidades de instalación y configuración en cada máquina cliente.

- Al ser una aplicación web, el usuario puede acceder a la misma desde cualquier cliente con conexión a Internet y un navegador, no siendo necesario ningún conocimiento adicional para el acceso a ella.

7.2 Trabajos futuros propuestos.

A lo largo del proceso de creación de *AWARD^{Potentials}* se han ido planteando diversas ideas y nuevas funcionalidades que harían mucho más completo el diseño inicial, pero que no han podido ser implementadas, bien por salirse del planteamiento inicial, o bien por la complejidad de implantación que aportaban al proyecto una vez superadas determinadas fases del mismo.

Pese a ello, consideramos interesante incluir un pequeño resumen sobre futuras mejoras a la aplicación que harán posible un mayor control sobre el seguimiento de los sujetos y ampliación de sus funcionalidades, básicas o avanzadas. A continuación mostraremos algunos de los posibles trabajos que se pueden realizar en base al proyecto finalizado:

- Unificación de las bases de datos de los proyectos AWARD. La idea sería incluir las tablas de la base de datos de *AWARD^{Potentials}* en la base de datos ya en uso del proyecto *AWARD^{Subject}* para así poder reutilizar la información de los sujetos ya estudiados en fases previas del proyecto AWARD para ir complementándola con nuevos datos de interés terapéutico. En la Figura 7.1 se puede observar la estructura actual de la base de datos de *AWARD^{Subject}*, destacando las tablas coincidentes entre los distintos proyectos.
- Implementar un visualizador de test histórico con el que observar la mejora evolutiva del sujeto. Este visualizador mostraría la disminución o aumento de tiempo en la aparición de determinados componentes (P100, P300,...), indicando si el sujeto está evolucionando favorablemente en los test realizados.
- Aumentar el uso de funcionalidades en la edición de test, tanto individuales como múltiples, respondiendo a las necesidades del terapeuta. Estos requerimientos deben ir apareciendo a medida que la herramienta vaya siendo usada por los profesionales de la neurofisiología y la neuropsicología.
- Implementar un módulo inteligente de asignado de etiquetas basado en un modelo de representación temporal sobre datos normativos.

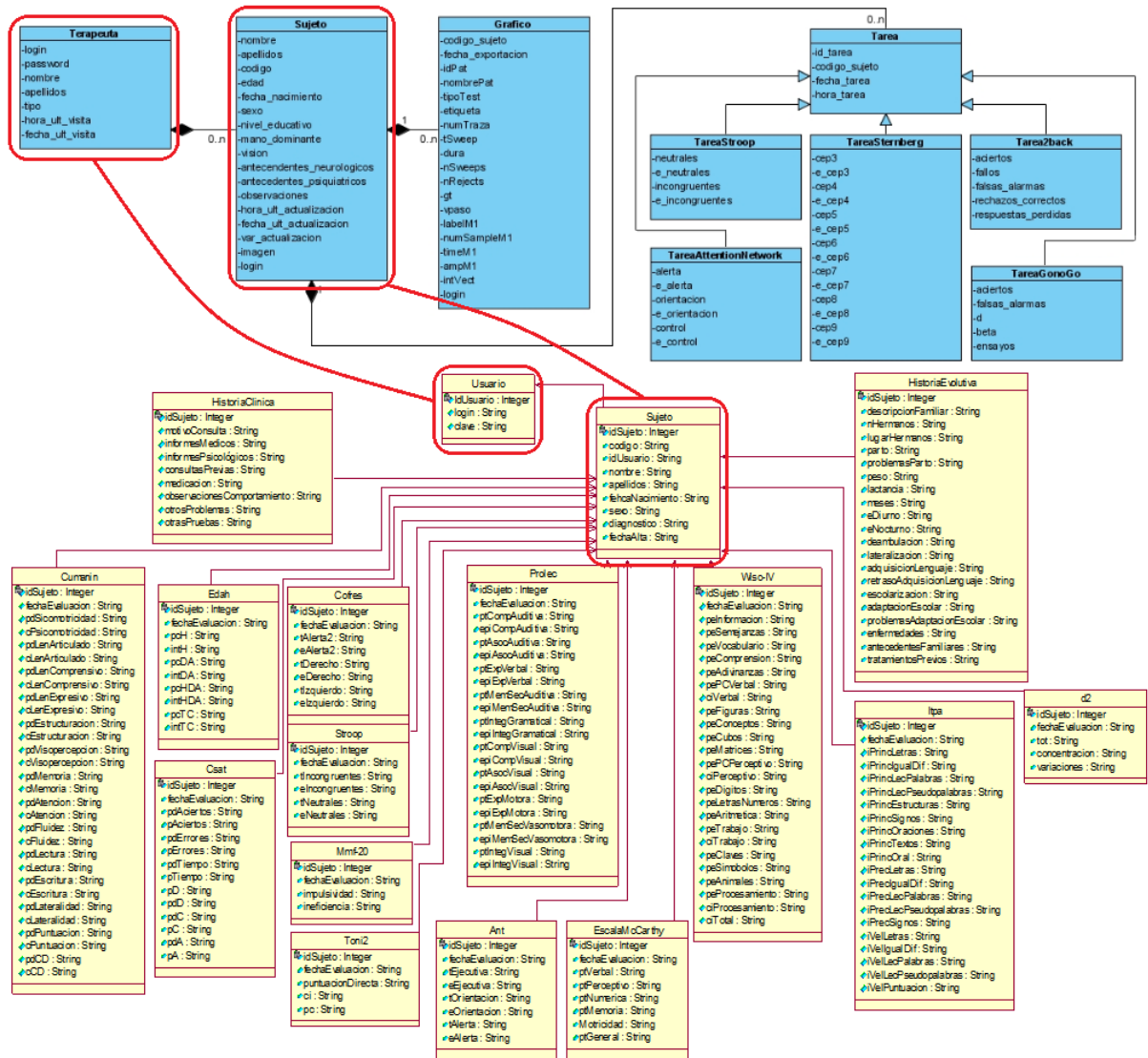


Figura 7.1: Unificación con la base de datos de AWARD^{Subject}.



Manual de instalación y usuario de *AWARD^{Potentials}*

En este capítulo se explicarán los pasos necesarios para instalar y usar *AWARD^{Potentials}*.

A.1 Instalación de software.

A.1.1. Instalación de software previo.

Para empezar, deberemos instalar todas las aplicaciones básicas para hacer funcionar *AWARD^{Potentials}* en nuestro ordenador. Estas aplicaciones/herramientas son PHP, MySQL y el servidor web Apache.

Todo este software se puede instalar de forma independiente, pero en este caso lo haremos de forma conjunta mediante un paquete llamado XAMPP que dispone de los siguientes recursos: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System para Win32 y NetWare Systems v3.32, JpGraph, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, y WEB-DAV + mod_auth_mysql.

Para instalar XAMPP sólo se requiere su descarga y extracción. No hay cambios en el registro de Windows (a no ser que se utilice la versión de XAMPP para el instalador de Windows) y no es necesario editar ningún archivo de configuración. En la Figura A.1 se muestra la página inicial de la aplicación.

La filosofía que ha seguido el desarrollo de XAMPP consiste en la construcción de una versión fácil de instalar para los desarrolladores que entran al mundo de Apache. XAMPP está configurado con todas las funciones activadas. La configuración por defecto no es buena desde el punto de vista de seguridad, por lo que no es suficientemente segura para un ambiente de producción.

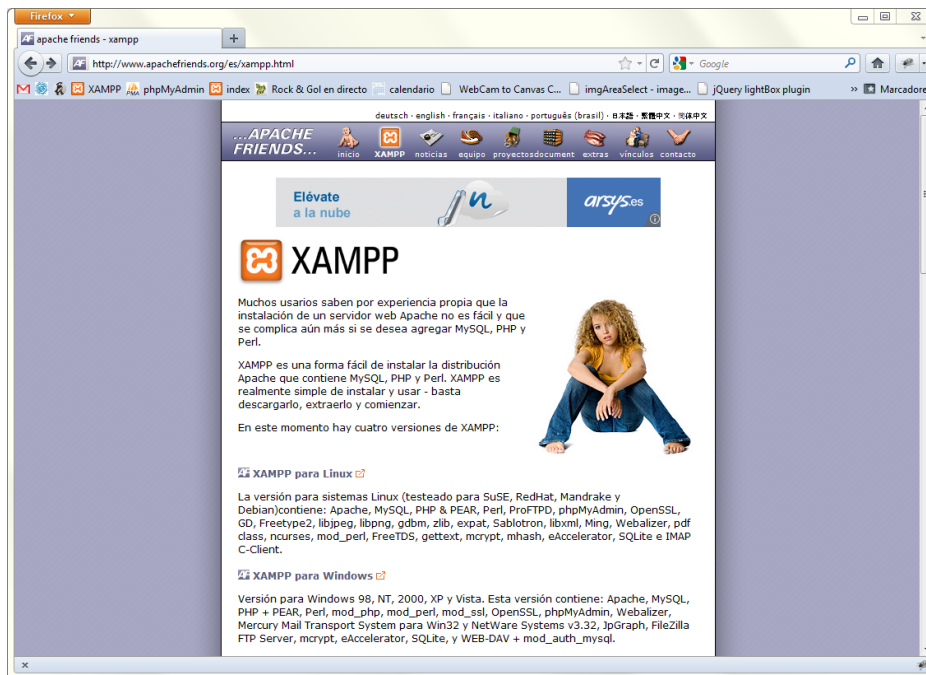


Figura A.1: Sitio web de XAMPP.

La versión de XAMPP que hemos utilizado ha sido la 1.7.4 (en la Figura A.2 se muestra la página de descarga), que disponía de las siguientes utilidades necesarias para el funcionamiento de *AWARD*^{Potentials}: Apache 2.2.17, MySQL 5.5.8, PHP 5.3.5 y phpMyAdmin 3.3.9.

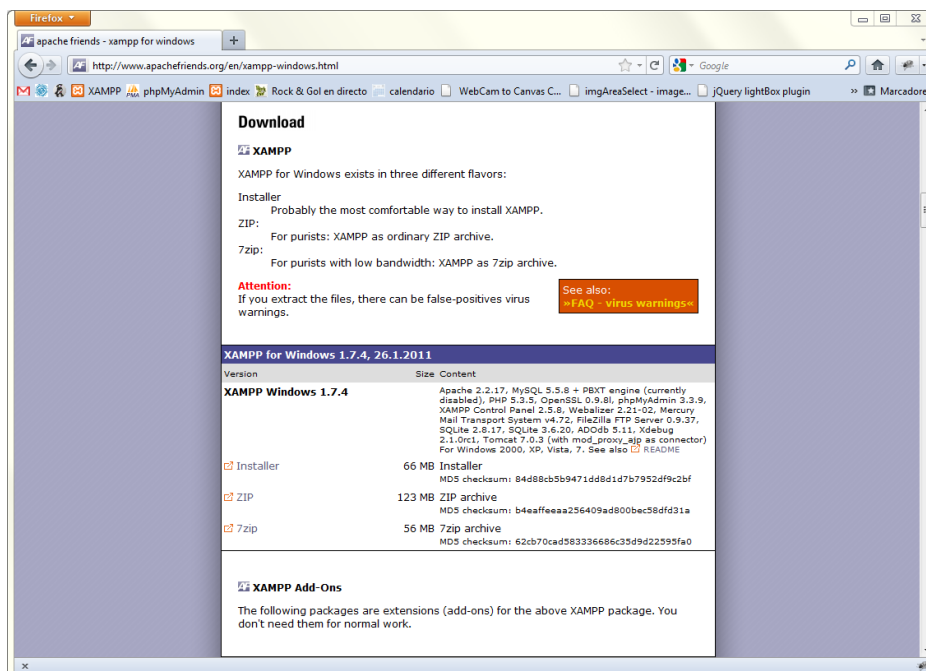


Figura A.2: Descarga de XAMPP para Windows.

Una vez descargado, la instalación es muy sencilla (véase Figura A.3).

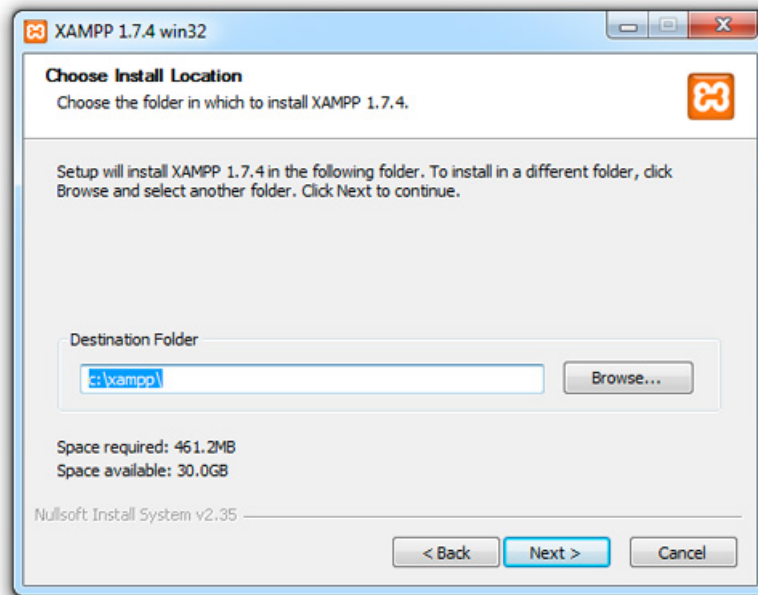


Figura A.3: Ventana de instalación de XAMPP.

Cuando instalación ha terminado, podremos encontrar la ruta del programa en Inicio/Programas/XAMPP y podremos utilizar el Panel de Control para iniciar y parar todos los servidores y poder también instalar y desinstalar servicios (ver Figura A.4).

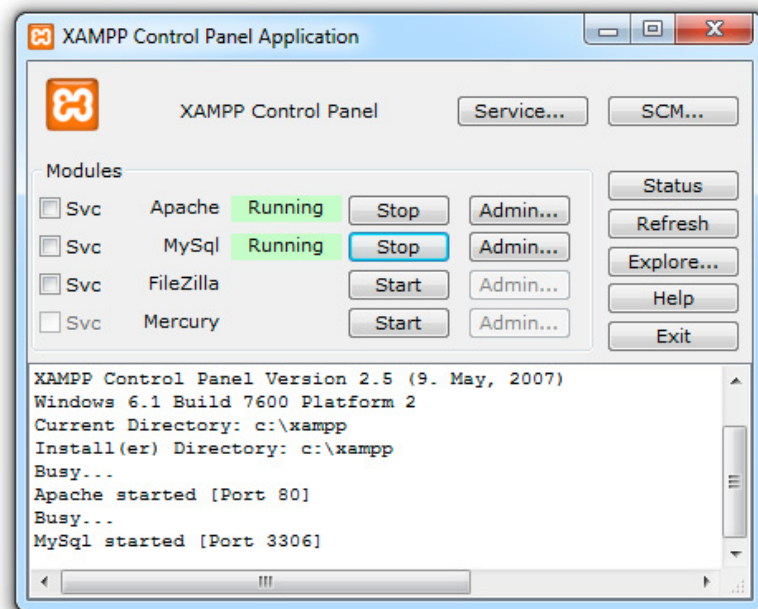


Figura A.4: Panel de control de XAMPP.

También podemos iniciar y parar todos los servidores de forma particular mediante estos archivos ejecutables:

- Apache y MySQL start: /xampp/xampp_start.exe
- Apache y MySQL stop: /xampp/xampp_stop.exe
- Apache start: /xampp/apache_start.bat
- Apache stop: /xampp/apache_stop.bat
- MySQL start: /xampp/mysql_start.bat
- MySQL stop: /xampp/mysql_stop.bat

Una vez iniciados los servidores de Apache y MySQL accederemos a la dirección <http://localhost/> o, lo que es lo mismo, <http://127.0.0.1/> mediante cualquier navegador mostrándose la pantalla de la Figura A.5 con ejemplos y herramientas.

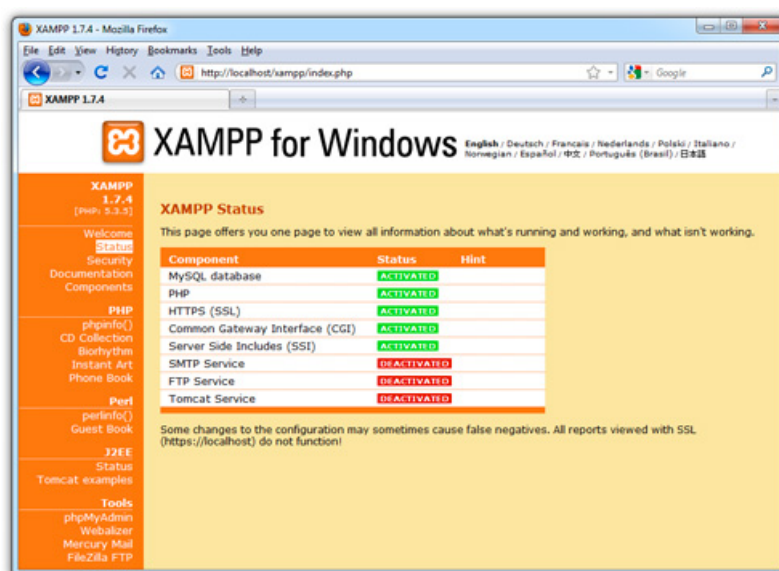


Figura A.5: Página de estado de XAMPP.

Como se ha mencionado anteriormente, XAMPP no es para un uso en producción, sino que es una utilidad para desarrolladores. XAMPP está configurado de la forma más abierta posible para permitir que el desarrollador web pueda hacer lo que él quiera. Para entornos de desarrollo es una herramienta muy potente, pero en un entorno de producción profesional podría ser mala idea su utilización.

A continuación se muestra una lista de opciones de seguridad que no se tienen en cuenta en XAMPP:

- El administrador de MySQL (root) no tiene contraseña.
- El demonio (daemon) de MySQL es accesible a través de la red.
- phpMyAdmin es accesible a través de la red.

- La página demo de XAMPP es accesible a través de la red.

Todos estos puntos pueden conllevar un riesgo de seguridad enorme. Especialmente si hacemos que XAMPP sea accesible a través de la red por personas fuera de la LAN. Para solucionar este problema, se puede hacer uso de un firewall o un router (NAT), ya que en este caso, se podría conseguir que el PC no fuese accesible a través de la red. En cualquier caso, es función del desarrollador o administrador del sistema solucionar estos problemas de seguridad. Como una pequeña ayuda podemos utilizar la “Consola de Seguridad de XAMPP”. Si queremos configurar XAMPP para que se pueda acceder desde Internet, deberemos ir a la URI <http://localhost/security/> para solucionar algunos de los problemas mencionados (ver Figura A.6).



Figura A.6: Página de seguridad de XAMPP.

Con la consola de seguridad se puede establecer una contraseña para el usuario MySQL “root” y phpMyAdmin. También puede activarse un sistema de autenticación para las páginas de demostraciones de XAMPP (ver Figura A.7).

A.1.2. Instalación de *AWARD Potentials*.

Una vez instalado todo el entorno de trabajo, solo necesitaremos instalar *AWARD Potentials*. Para ello necesitaremos instalar y configurar en primer lugar la base de datos MySQL así como copiar el código de la aplicación en la ruta donde apunte el servidor apache.

Para comenzar, importaremos el archivo .sql de la base de datos de *AWARD Potentials* desde el entorno de PHPMyAdmin (ver Figura ??). Únicamente deberemos entrar en la URI <http://localhost/phpmyadmin/> y, a continuación, dirigirnos al apartado *Importar*. En la localización del archivo de texto buscaremos el archivo SQL de nuestra base de



Figura A.7: Página de seguridad de XAMPP, cambio de contraseña.

datos y marcaremos el checkbox del tipo de archivo importado. Después de hacer clic en continuar, la base de datos se encontrará almacenada en nuestro sistema.

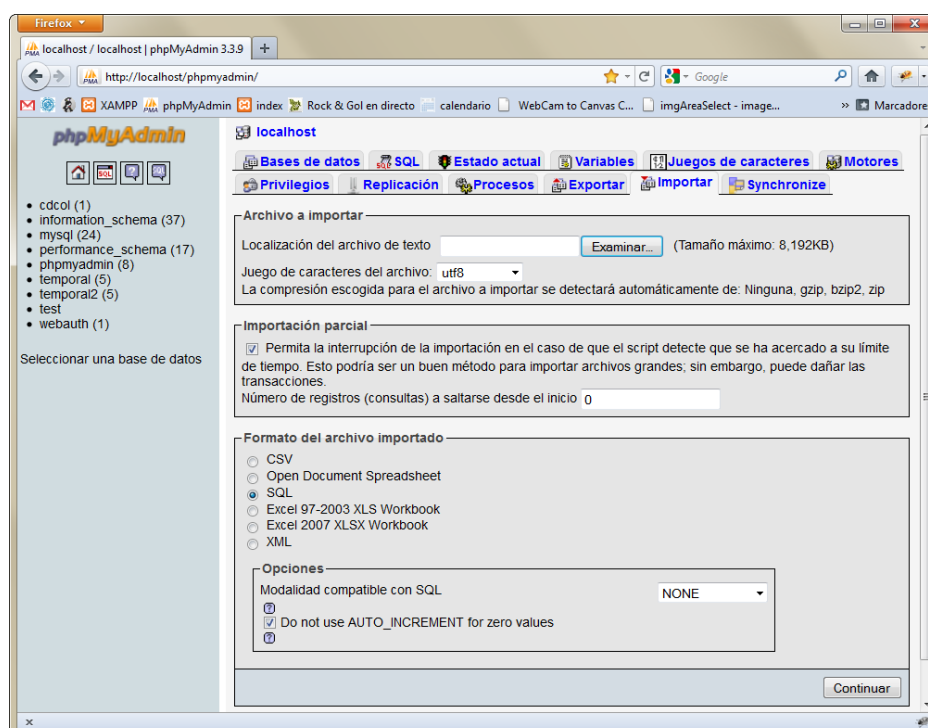


Figura A.8: Importación de base de datos SQL desde PHPMyAdmin.

Una vez instalada la base de datos, únicamente deberemos copiar los archivos fuente de AWARD^{Potentials} en el directorio donde apunta el servidor Apache (ver Figura A.9). Al

haber instalado Apache mediante el software XAMPP, este directorio deberá encontrarse en la ruta C:/xampp/htdocs.

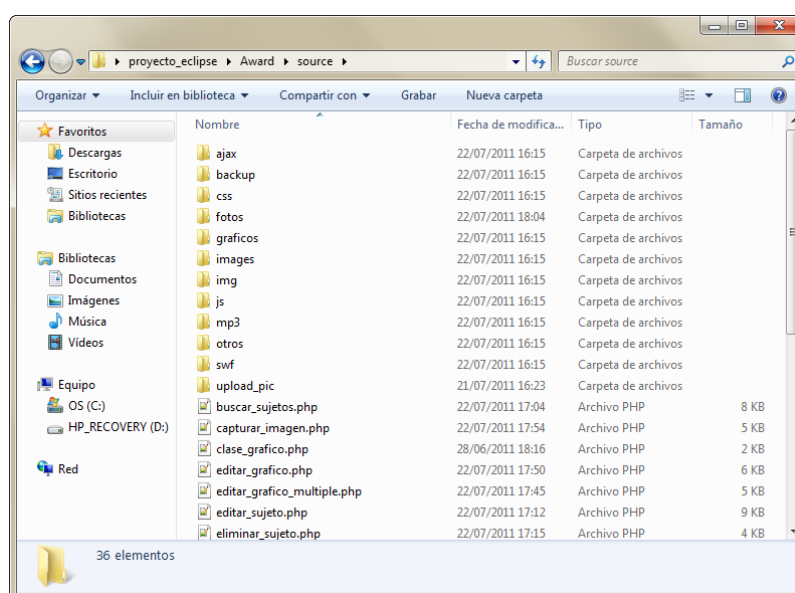


Figura A.9: Archivos fuente de *AWARD^{Potentials}*.

A.2 Manual de usuario.

A continuación se presentará un breve manual de usuario de la aplicación *AWARD^{Potentials}*, mostrando en cada momento los pasos a seguir por el terapeuta para realizar cualquier función implementada en el sistema.

A.2.1. Sesión del terapeuta.

El terapeuta podrá acceder a la aplicación mediante cualquier navegador, encontrándose la página inicial que se muestra en la Figura A.10. Esta pantalla nos muestra algo de información sobre el proyecto AWARD y nos permite la opción de entrar en el sistema mediante un *login* y un *password*.

Una vez que se accede al sistema, el terapeuta podrá ver una pantalla informativa (ver Figura A.11) con información acerca de los sujetos que él mismo dió de alta. Los sujetos que aparecen en esta pantalla son los últimos que fueron modificados por el terapeuta. Del mismo modo, en la parte izquierda de la pantalla se mostrarán los últimos gráficos modificados por el terapeuta.

Si se accede a la página de búsqueda de sujetos, el terapeuta tendrá dos opciones de búsqueda (ver Figura A.12). La búsqueda básica le permite buscar sujetos unicamente por nombre y apellido, mientras que con la búsqueda avanzada, el terapeuta podrá buscar un sujeto mediante cualquier campo perteneciente a éste (género, fecha de nacimiento, nivel



Figura A.10: Captura de pantalla index.php.



Figura A.11: Captura de pantalla terapeuta.php.

educativo, etc.). Una vez realizada la búsqueda, se mostrará una lista con todos los sujetos coincidentes con los parámetros de búsqueda indicados.

El terapeuta también podrá dar de alta en el sistema un nuevo sujeto desde la página

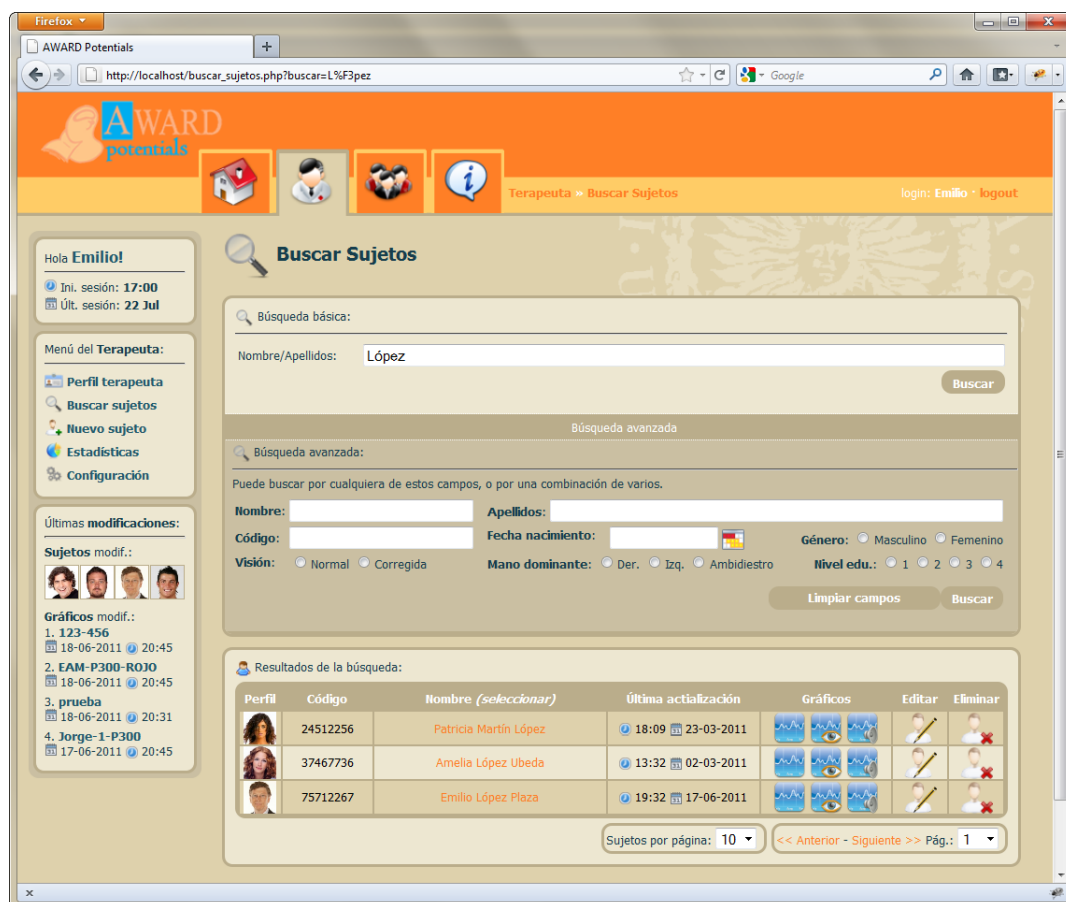


Figura A.12: Captura de pantalla buscar_sujetos.php.

mostrada en la Figura A.13. En esta sección se muestra un formulario con todos los campos necesarios para dar de alta un nuevo sujeto. Hay que tener en cuenta que el código de sujeto debe de ser único, por lo que si el terapeuta introduce un código ya existente en la base de datos, la aplicación nos avisará en ese mismo instante del error y no nos permitirá guardar los cambios en la base de datos.

En la parte de la derecha de la pantalla hay un menú desplegable en la que se indica que podemos elegir un servidor para importar información de sujetos. Esta funcionalidad no se encuentra implementada, ya que se pensó como trabajo futuro para utilizar la aplicación con sujetos de otros proyectos AWARD.

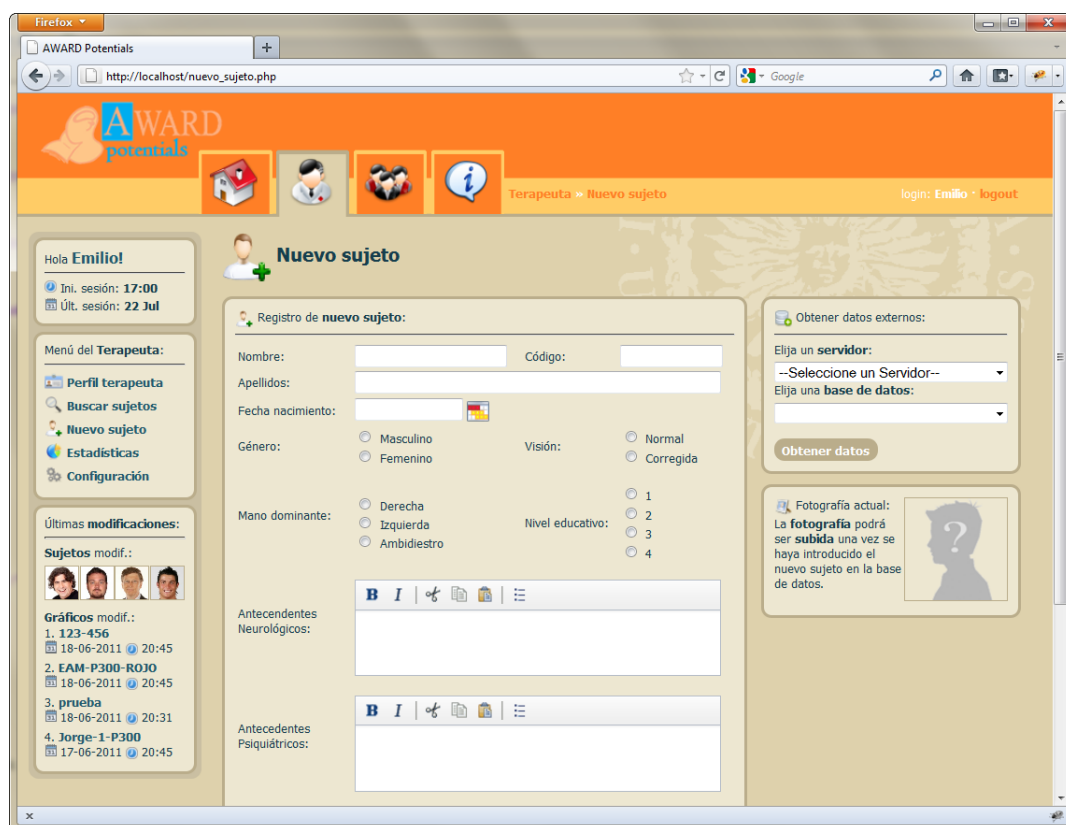


Figura A.13: Captura de pantalla nuevo_sujeto.php.

A.2.2. Gestión de sujetos.

Al acceder a la sección de sujetos (ver Figura A.14), el terapeuta podrá ver un listado con todos los sujetos que él mismo dio de alta en el sistema. Desde aquí el terapeuta podrá seleccionar un sujeto haciendo clic sobre el nombre (se mostrará su nombre y fotografía en la parte izquierda de la pantalla) y, a continuación, realizar cualquier acción que se desee con él (ver/modificar perfil, visualizar/editar gráficos, eliminarlo del sistema, etc).

The screenshot shows a web browser window with the URL `http://localhost/sujetos.php?codigo=43713381`. The page title is "Listado de sujetos" and the user is logged in as "Emilio". The main content area displays a table of subjects:

Perfil	Código	Nombre (seleccionar)	Última actualización	Gráficos	Editar	Eliminar
	666	Jorge Cano Fernández	18:49 16-06-2011			
	23232	Cristiano Ronaldo Martínez	15:29 29-04-2011			
	1111111	Pepe Cespedes	16:30 22-07-2011			
	24512256	Patricia Martín López	18:09 23-03-2011			
	26378816	Chuck Norris Padilla	13:46 10-03-2011			
	34512267	Emilio Pérez Lasa	14:41 27-04-2011			
	36256617	Jose Luis Torrente Segura	11:27 15-03-2011			
	36726615	Andrés Iniesta Segura	13:18 02-03-2011			
	36726616	Alicia Pelaez Marchena	20:52 20-02-2011			
	37467736	Amelia López Ubeda	13:32 02-03-2011			
	43713381	Emilio López Plaza	17:09 22-07-2011			
	47895587	Raimundo Amador	13:34 02-03-2011			
	53128816	Jose Antonio Martín Padilla	12:14 2-03-2011			
	74637728	María del Mar Muñoz Alarcón	11:12 23-01-2011			

Figura A.14: Captura de pantalla sujetos.php.

Si el terapeuta accede a la página del perfil del sujeto (ver Figura A.15) podrá ver todos los datos del sujeto que se encuentran en la base de datos. Además, en la parte de la derecha tenemos un formulario para la importación de gráficos. Desde aquí se podrá seleccionar un gráfico exportado por la aplicación MaxG EP-01 y asignárselo al sujeto que queramos (deberemos elegir previamente si el gráfico que vamos a subir es de tipo visual o auditivo).

Si el terapeuta desea eliminar alguno de los sujetos que tiene asignados, podrá hacerlo desde la página de eliminación de sujetos (ver Figura A.16). A esta página se accede desde el listado de sujetos que hemos visto anteriormente. La aplicación nos pedirá una confirmación para la eliminación del sujeto seleccionado y, una vez obtenida, se eliminará al sujeto junto con todos los gráficos asignados a éste.

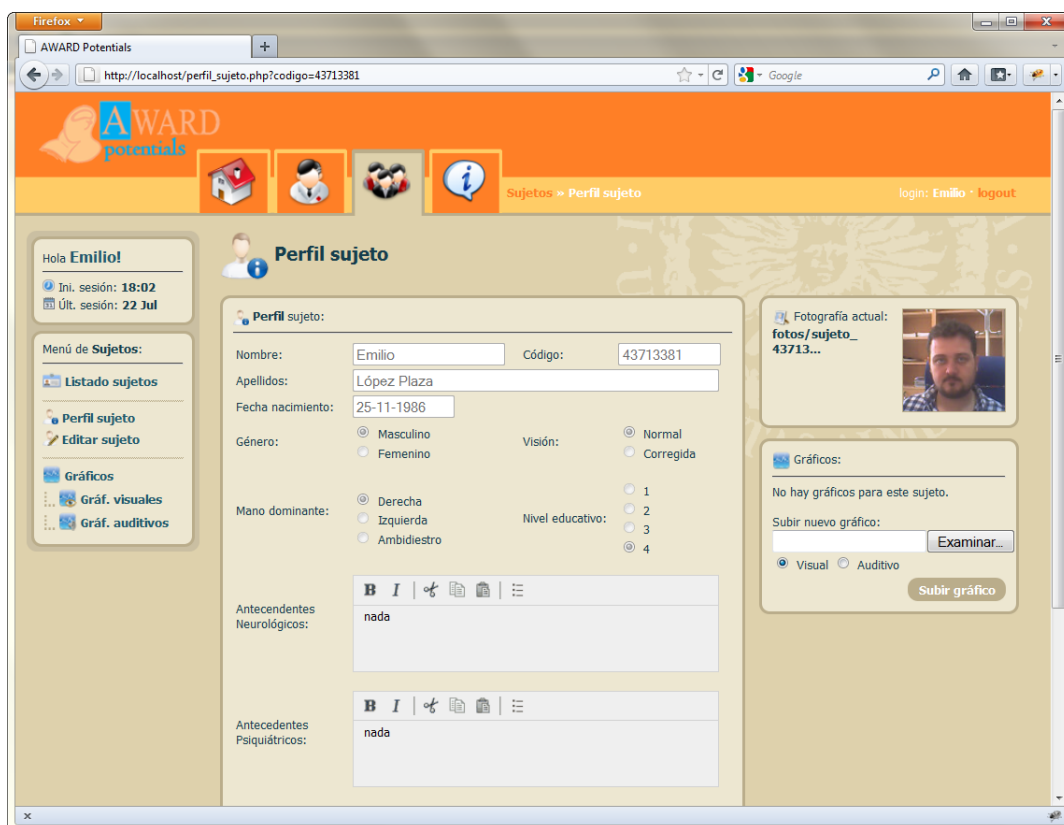


Figura A.15: Captura de pantalla perfil.sujeto.php.

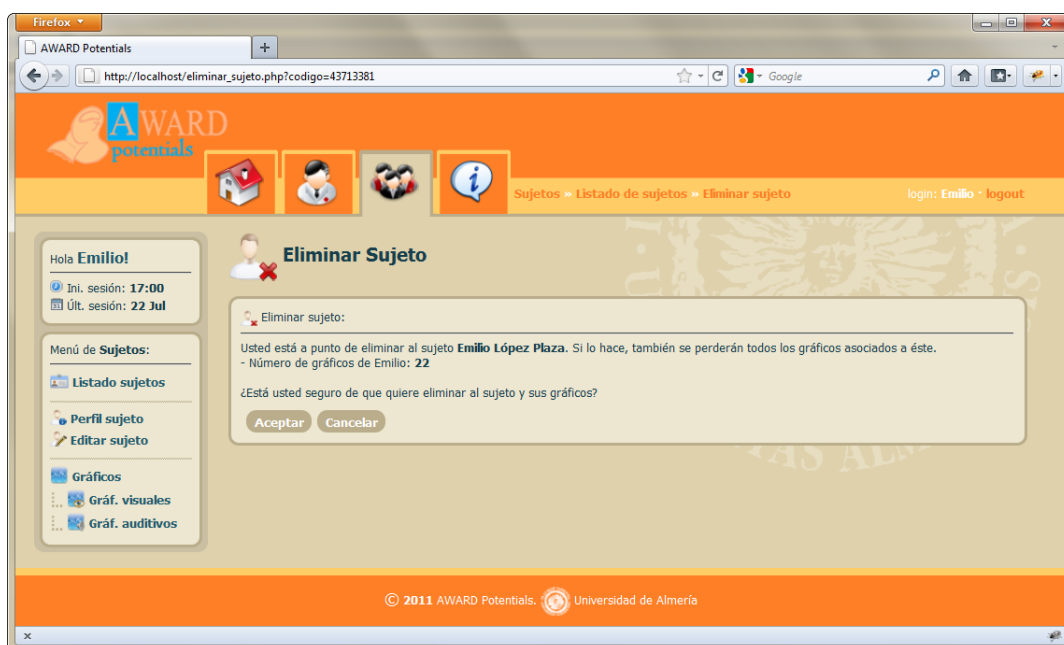


Figura A.16: Captura de pantalla eliminar_sujeto.php.

El panel de edición de sujetos (ver Figura A.17) es muy similar al de perfil de sujetos aunque, en este caso, el terapeuta podrá modificar los campos que desee y volver a guardarlos en la base de datos. Desde esta página el terapeuta podrá actualizar la fo-

tografía del sujeto (subiendo un archivo de imagen válido o bien capturando la imagen a través de la webcam) y subir nuevos gráficos al igual que en la página de perfil vista anteriormente.

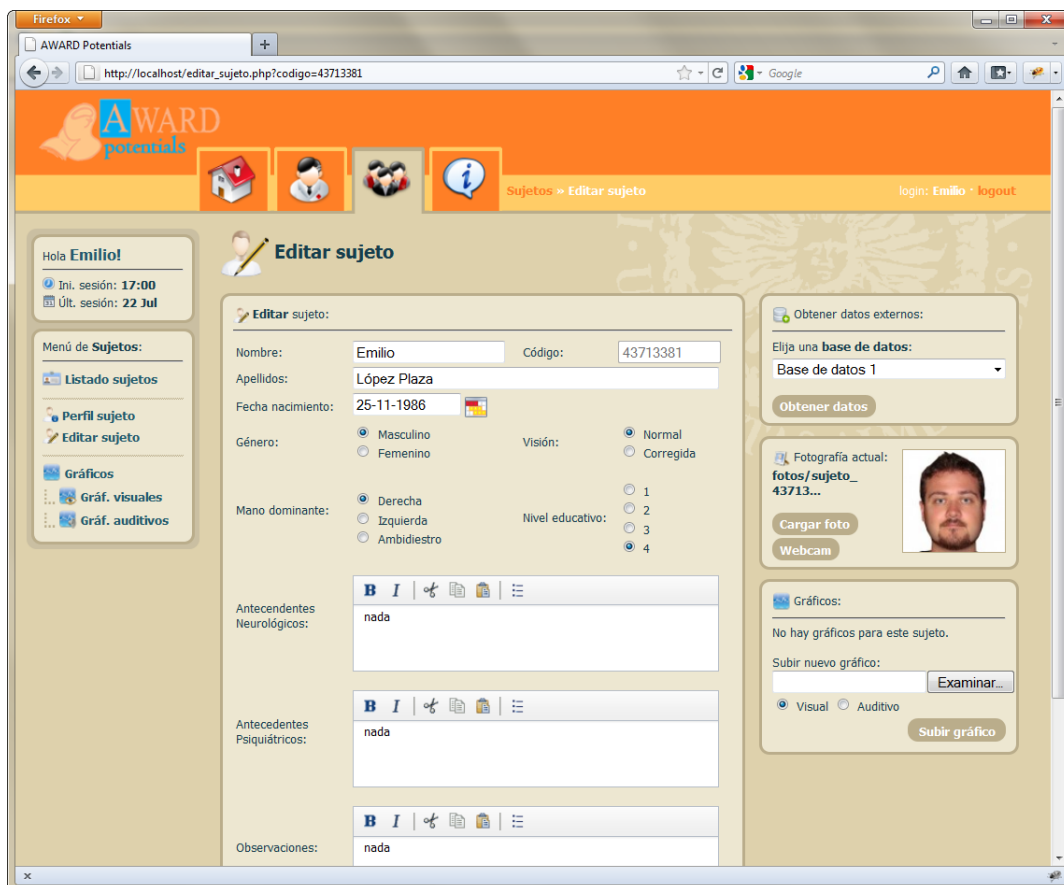


Figura A.17: Captura de pantalla editar_sujeto.php.

Como se ha comentado anteriormente, desde la página de edición de sujeto se puede subir una nueva imagen personal y asignársela. En la Figura A.18 se muestra el formulario desde el cual se puede seleccionar un archivo de imagen local.

Una vez seleccionado el archivo de imagen, éste se subirá al sistema y lo visualizará (ver Figura A.19). En este momento, el terapeuta podrá seleccionar el área de la fotografía que desee (normalmente el área debería abarcar la cara del sujeto) y recortarla para su almacenamiento posterior en la base de datos. Una vez seleccionada el área, se podrá hacer clic en guardar selección para introducir la imagen en la base de datos.

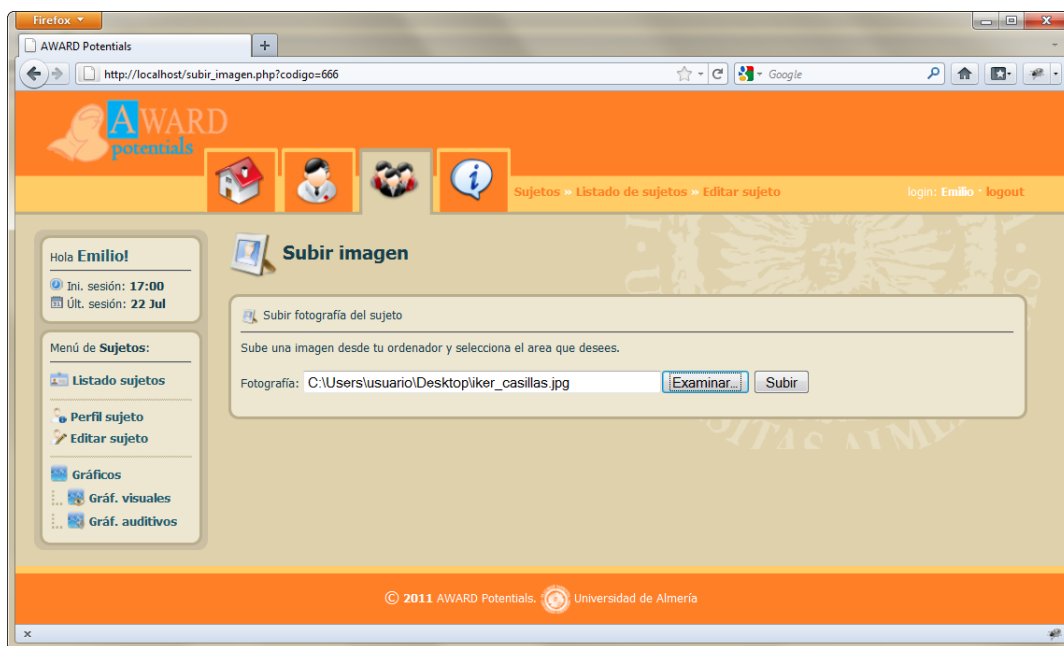


Figura A.18: Captura de pantalla subir_imagen.php.

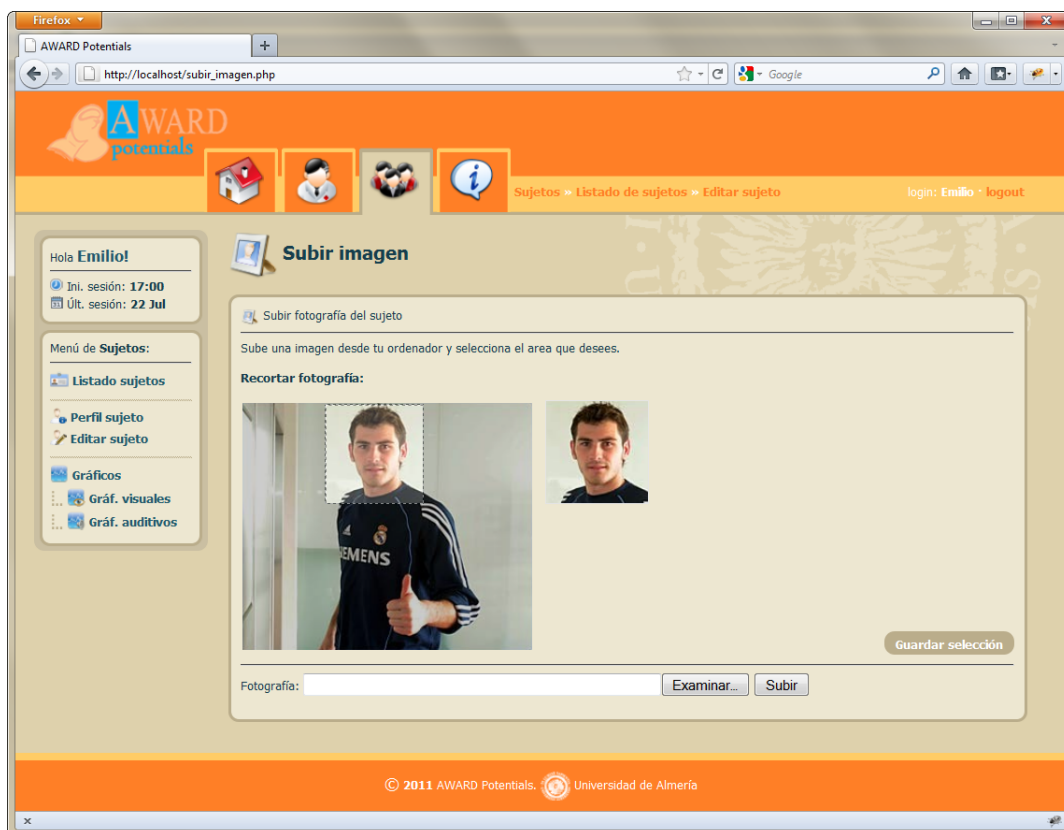


Figura A.19: Captura de pantalla subir_imagen.php (2).

La otra forma de asignar una imagen a un sujeto es capturándola en tiempo real mediante webcam (ver Figura A.20). En esta página se muestra la imagen capturada por

una webcam conectada al puerto USB mediante una pequeña aplicación Flash. Una vez capturada la imagen, se podrá guardar en el sistema e introducir su ruta en la base de datos al igual que en el caso anterior.

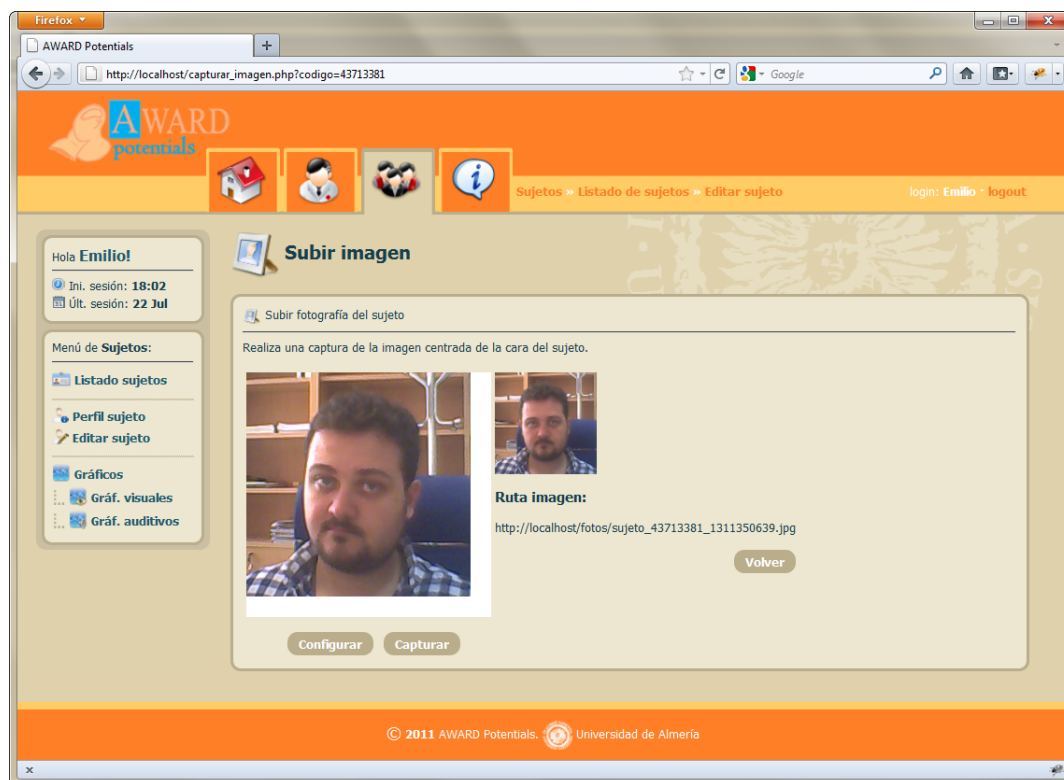


Figura A.20: Captura de pantalla capturar_imagen.php.

A.2.3. Gestión de gráficos (potenciales evocados).

Desde el listado de sujetos, el terapeuta podrá dirigirse a la pantalla de gestión de gráficos (ver Figura A.21). El terapeuta podrá elegir si quiere visualizar los gráficos visuales, auditivos o todos los gráficos pertenecientes al sujeto en cuestión.

Una vez situado en la pantalla de gestión de gráficos, el terapeuta encontrará (en la zona superior) información acerca del sujeto al que pertenecen los gráficos, un listado con los últimos gráficos modificados para ese sujeto y, además, un formulario donde subir nuevos gráficos desde esta misma pantalla.

A continuación se mostrarán todos los gráficos del sujeto ordenados por orden decreciente de su fecha de modificación. En la esquina superior derecha de cada gráfico hay tres iconos. El primero (con forma de símbolo “+”) tiene al lado un componente *checkbox*, el cual podrá ser marcado en uno o más gráficos para después combinarlos y editarlos conjuntamente. El segundo (con forma de ojo) nos mostrará todos los campos con los datos del gráfico que se almacenan en la base de datos. Y, por último, el tercer icono (con forma de engranaje) nos permitirá editar el gráfico de forma individual. Al final de la pantalla se encuentra el botón “Combinar” con el que podremos realizar una edición

múltiple de los gráficos cuyo componente *checkbox* hemos marcado anteriormente.



Figura A.21: Captura de pantalla graficos.php.

En la Figura A.22 se puede observar la información de la página grafico.php (a la cual

se puede acceder desde la pantalla de gestión de gráficos, o bien mediante el listado de gráficos del sujeto), la cual contiene todos los campos almacenados en la base de datos para cada uno de los gráficos de cada sujeto. Estos campos no son editables, ya que se almacenan automáticamente al importar un archivo exportado del EP-01 a nuestra aplicación.

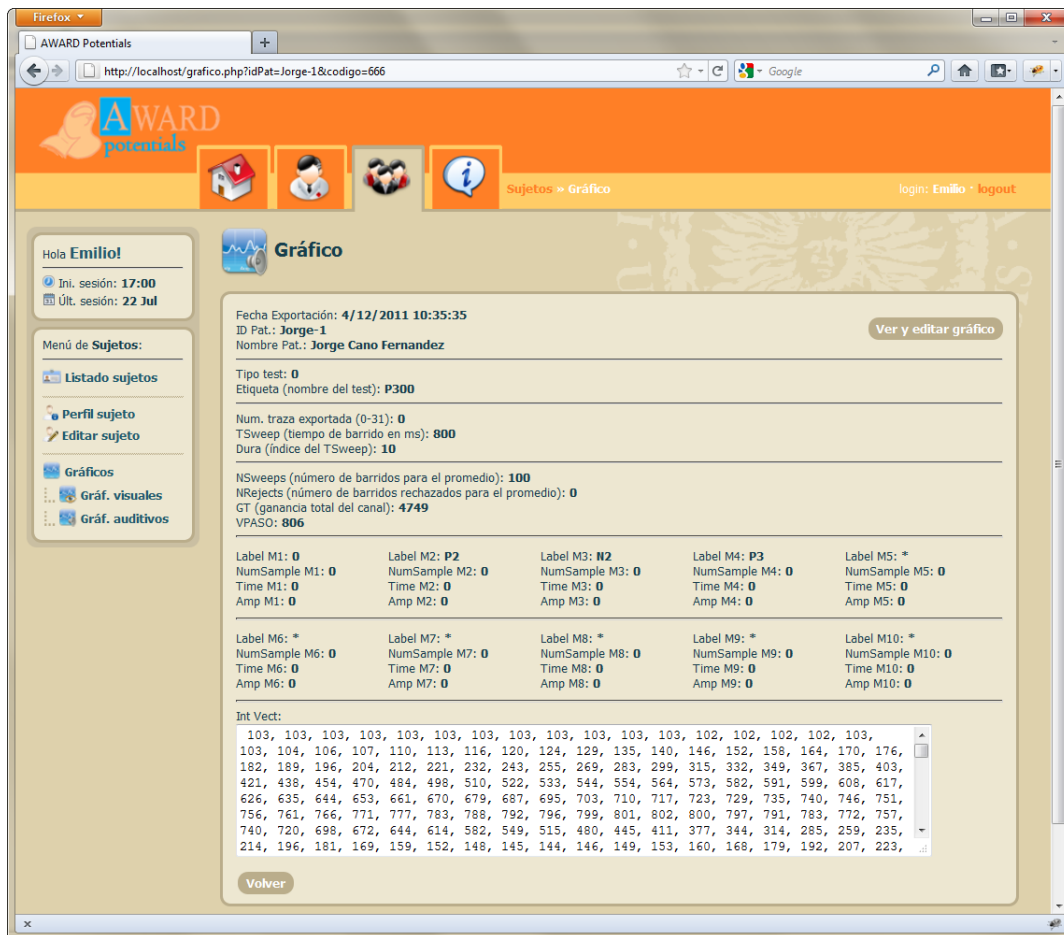


Figura A.22: Captura de pantalla grafico.php.

Como se ha comentado anteriormente, los gráficos pertenecientes a un determinado sujeto se pueden visualizar y editar de forma individual o conjunta. En la Figura A.23 se puede observar el panel de edición de gráficos individuales.

En la parte superior de este panel se encuentra información relativa tanto al sujeto al que pertenece el potencial evocado que se está editando, como al terapeuta al que pertenece ese sujeto. También disponemos de dos botones que nos permiten volver al perfil del sujeto y otro que nos redirige a la página de información del potencial evocado en la que encontraremos toda la información almacenada en la base de datos. Además, el terapeuta podrá realizar una serie de actividades orientadas al estudio de los potenciales evocados. A continuación, se explicarán las funciones más representativas del editor gráfico simple:

- *Inserción de etiquetas.* El terapeuta podrá insertar etiquetas a lo largo de la curva

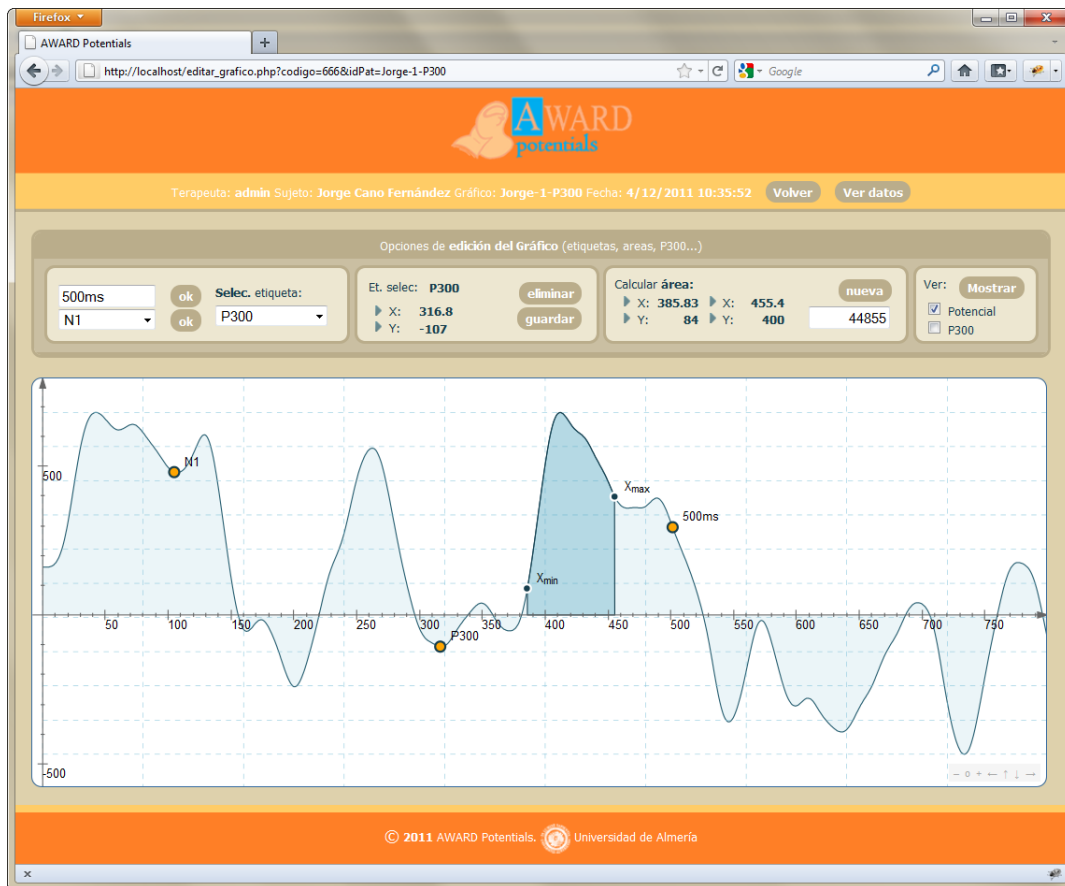


Figura A.23: Captura de pantalla editar_grafico.php.

del potencial. Esta inserción se podrá realizar de dos formas. La primera consiste en escribir en el primer *input* de texto el nombre de la etiqueta que se desee y, una vez pulsado el botón “Ok”, aparecerá al principio del gráfico para que el terapeuta pueda moverlo y posicionarlo con el ratón sobre la propia curva en el punto que desee. La segunda forma consiste en seleccionar a partir de un menú desplegable alguno de los componentes típicos de los potenciales evocados (tras esto, la forma de actuación sería igual que en el caso anterior). En la figura anterior se puede observar que existe un menú desplegable con todas las etiquetas insertadas hasta el momento para así poder trabajar con ellas de forma individual.

- *Eliminación y almacenamiento de etiquetas en la base de datos.* En el segundo recuadro de funciones del panel de edición simple se puede observar los botones de “eliminar” y “guardar”. El botón de eliminar funciona habiendo seleccionado previamente la etiqueta deseada en el menú desplegable. Una vez seleccionada y habiendo pulsado el botón de eliminar, dejará de representarse en el gráfico dicha etiqueta y, además, si estaba almacenada en la base de datos, se lanzará una consulta para eliminarla completamente. El botón “guardar” almacena la etiqueta seleccionada en la base de datos.
- *Cálculo de áreas.* En el tercer recuadro del menú del panel de edición simple encontramos la opción de insertar área. Para utilizar esta funcionalidad únicamente

deberemos pulsar en el botón “nueva” y aparecerá dibujada en la curva del potencial evocado. Una vez representada veremos dos puntos (gliders) denominados X_{min} y X_{max} , los cuales se podrán mover desde el mismo panel para modificar el comienzo y fin del área representada. Mientras movemos estos puntos se irá actualizando automáticamente el panel informativo en el que se muestra el área total generada junto con las coordenadas de los puntos mencionados anteriormente.

- *Visualización.* El último recuadro del menú nos permite seleccionar qué tipo de gráfico o combinación de ellos deseamos visualizar en el panel de edición. Las opciones de las que disponemos son: (1) Visualizar el Potencial Evocado (visual o auditivo); (2) visualizar el gráfico no frecuente (P300) relativo al potencial evocado seleccionado; y (3) visualizar tanto el potencial evocado como su P300 en el mismo gráfico (esta opción nos llevaría a una página de edición de gráfico múltiple con los dos gráficos representados).

Del mismo modo, en la Figura A.24 se puede observar el panel de edición de gráficos múltiples, el cual dispone de las siguientes funciones:

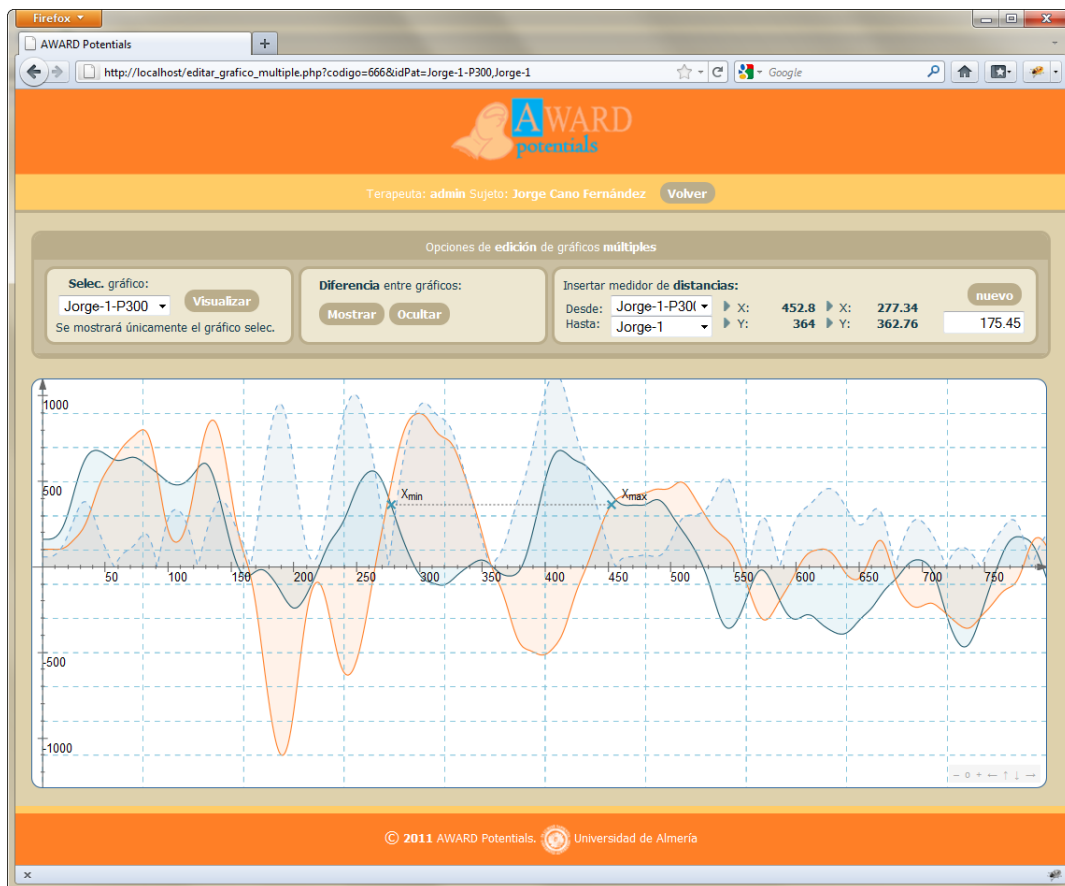


Figura A.24: Captura de pantalla editar_grafico_multiple.php.

- *Selección de gráfico.* Esta opción nos permite elegir alguno de los potenciales representados en ese momento y visualizarlo por separado mediante el panel de edición

simple. Antes, deberemos elegir el gráfico deseado en el menú desplegable que se encuentra en el primer recuadro.

- *Diferencia entre gráficos.* Pulsando el botón “Mostrar” podremos visualizar un nuevo gráfico representado con una línea de puntos discontinua que se calcula promediando el total de los potenciales visualizados en el panel. Esta función es bastante útil para el terapeuta si, por ejemplo, se quiere observar la diferencia real entre un potencial evocado y su P300 (o también entre varios potenciales obtenidos a un mismo sujeto en un corto espacio de tiempo para concluir si la señal es fiable o no). Pulsando el botón “Ocultar” podremos dejar de visualizar el gráfico de la diferencia obtenido.
- *Insertar medidor de distancias.* Antes de realizar esta operación debemos seleccionar en los dos menús desplegables, denominados “Desde” y “Hasta”, los potenciales evocados para los que deseamos medir la distancia. Pulsando en el botón “Nuevo” se dibujará sobre las curvas de los gráficos seleccionados una línea discontinua que podremos mover pulsando sobre sus extremos y que se desplazará sobre la curva seleccionada en cada caso. Mientras movemos los extremos se irá recalculando y visualizando la distancia entre los dos puntos automáticamente.

B

Manual de usuario de *MaxG EP-01*

B.1 Aspecto inicial.

El aspecto de la pantalla del software *MaxG EP-01* es el que se muestra en la Figura B.1.

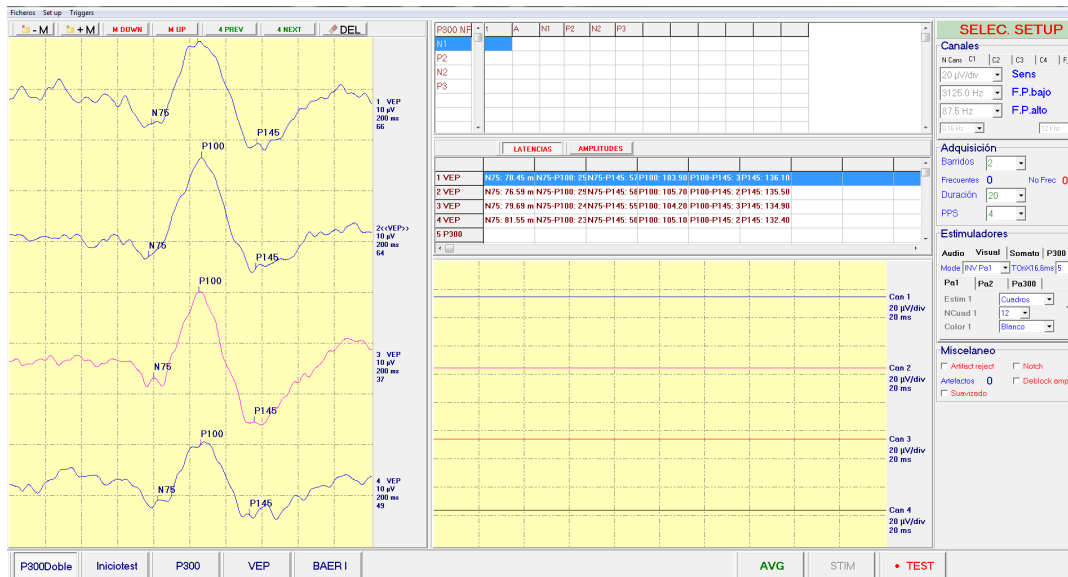


Figura B.1: Aspecto del software MaxG EP-01.

Antes de comenzar a trabajar con este software se deberán tener algunos factores en cuenta:

1. Conectar el módulo *EP-01* y el módulo *VISTIM* siempre al mismo conector USB del ordenador. Esto evita posibles redefiniciones de puertos serie.

2. Siempre se han de conectar los módulos antes de iniciar la aplicación.
3. Habrá que esperar a que el ordenador haya hecho el arranque al completo antes de iniciar la aplicación.

Una vez que se ha comprobado que todos los factores anteriores son correctos, se podrá comenzar con un tipo de test previamente almacenado (ver Figura B.2). Para ello:

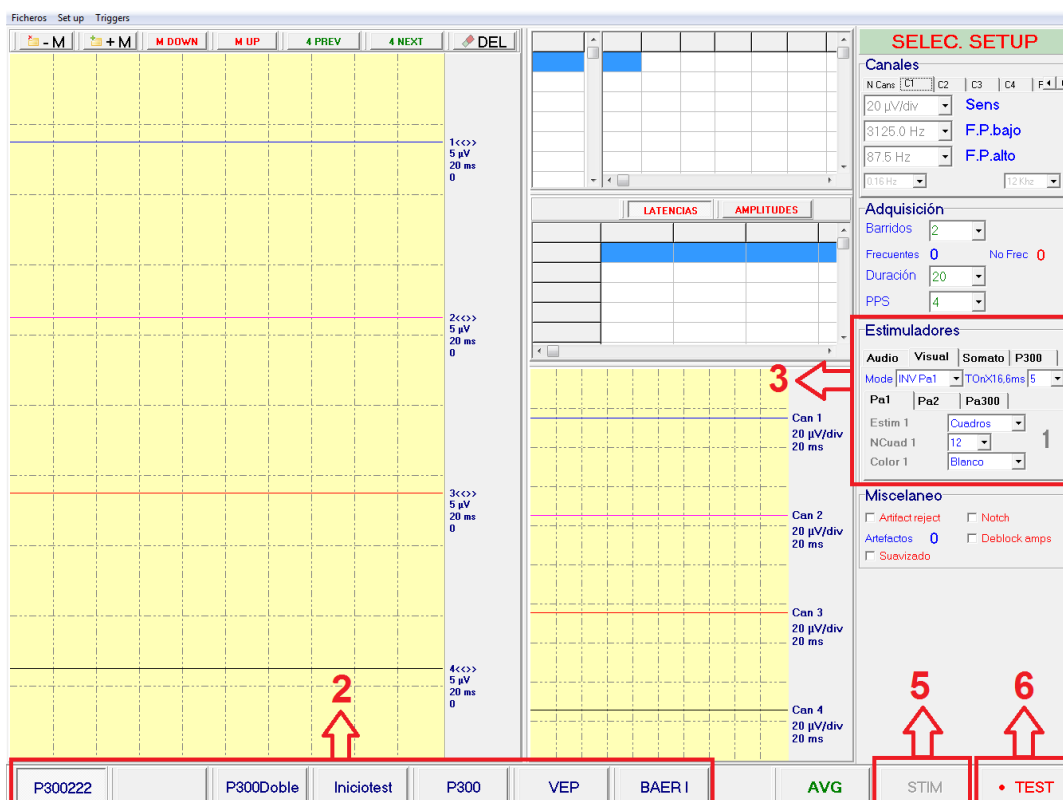


Figura B.2: Aspecto inicial de MaxG EP-01.

1. Entrar a la aplicación desde el acceso directo.
2. Hacer clic sobre el botón inferior del test elegido.
3. Actuar sobre el control del estimulador.
4. Colocar los electrodos sobre el sujeto.
5. Hacer clic sobre el botón de iniciar test. En este momento se habilitará el botón STIM.
6. Hacer clic sobre STIM para ver la calidad de la señal conseguida.
7. Retocar los electrodos hasta que la señal sea aceptable.
8. Informar al sujeto de lo que se espera de él y cómo debe comportarse cuando se le presenten los estímulos.

9. Hacer clic en AVG para comenzar con el promediado de la señal.
10. La aplicación se detendrá al llegar al número de barridos preprogramado.
11. Se podrá detener el estimulador con STIM cuando sea preciso.
12. El proceso de promediado se detiene si el botón AVG se desactiva, aunque el estimulador puede seguir funcionando.

B.2 Controles de memorias.

La aplicación consta de 32 memoria disponibles, numeradas de la 1 a la 32. Pueden ser utilizadas con tipos de test diferentes, aunque no es aconsejable por la complicación que puede añadir tal acción. Cada memoria contiene una suma de barridos de señal iguales en longitud temporal, tipo de test y amplificación hardware. Si se intenta añadir un barrido de señal distinto a una memoria que no esté vacía, el programa emitirá un señal sonora y se impedirá el promediado.

El programa se inicia siempre con la memoria M1 activa (notar el símbolo << a la derecha de la línea base), y muestra cuatro memorias por defecto.

- **-M**: Decrementa el número de memorias mostradas.
- **+M**: Incrementa el número de memorias mostradas.

Siempre se muestra un número de memorias de forma relativa a una determinada. Por ejemplo, si mostramos cuatro memorias a partir de la número 1, se podrán observar las memorias 1, 2, 3 y 4.

- **M DOWN**: Se incrementa el índice de la primera memoria mostrada, es decir, si estábamos viendo desde la 1, comenzamos a mostrar desde la 2, 3,...
- **M UP**: Comenzamos a ver memorias anteriores.

B.2.1. Elección de memoria de promediado.

El equipo puede recibir señal desde 1, 2 ó 4 canales simultáneamente. Cada cana necesita una memoria donde acumular barridos. Si el test tiene cuatro canales, necesitamos cuatro memorias contiguas.

- **1 NEXT**: Si hay un test que usa un sólo canal, este botón incrementa en uno el índice de la memoria de trabajo. Si hubiera cuatro canales activos en el test, el salto sería proporcional.
- **1 PREV**: Igual pero hacia atrás. Si las memorias que se encuentran en la posición de más atrás están ocupadas, sólo será permitido promediar sobre ellas en las mismas condiciones que contengan.

En caso del componente P300 se necesitan siempre el doble de memorias que canales usados. Así, para un test P300 con dos canales, necesitamos dos memorias para el resultado del estímulo frecuente, y dos más para el infrecuente. Se recomienda no activar un test P300 si por debajo de la memoria actual no queda el número suficiente de memorias libres.

B.3 Realización de tests.

A continuación se explicarán los pasos necesarios para crear un test desde cero, actualizar un test ya creado y borrar un test del sistema.

B.3.1. Crear un test nuevo.

Para crear un test desde cero se han de seguir los siguientes pasos:

1. Elegir las condiciones de adquisición (ver Figura B.3):
 - Número de canales.
 - Amplificación.
 - Filtrado.
 - Número de barridos.
 - Tiempo de análisis.
 - Pulsos por segundo.
2. Introducir las etiquetas de marcas que usará el test.
3. Hacer clic en SetUp (menú de la ventana principal, arriba)
4. Hacer clic en Crear SetUp.
5. Nombrar en la ventana que se abre y hacer clic en OK.
6. Hacer clic en SetUp/Salvar a disco. El ordenador emitirá dos bips rápidos.

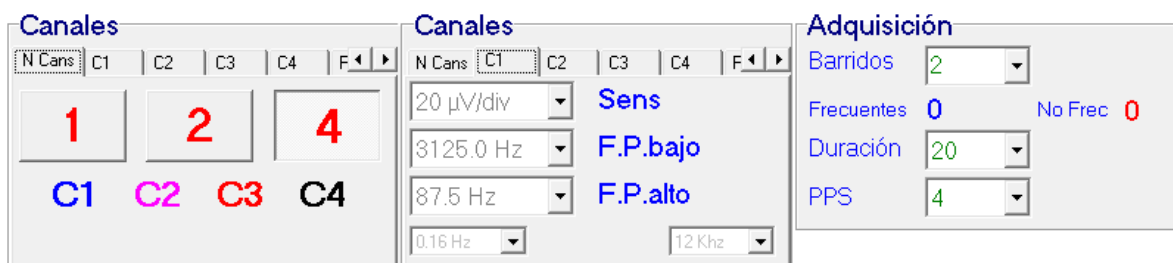


Figura B.3: Condiciones de adquisición.

El test nuevo estará disponible para ser aplicado, pero no puede actualizarse con nuevos parámetros hasta que no se reinicie la aplicación.

B.3.2. Actualizar un test ya creado.

Para actualizar un test ya creado se han de seguir los pasos que se muestran a continuación:

- Hacer clic sobre el botón test.
- Cambiar/editar los parámetros o etiquetas.
- Pulsar y mantener la tecla CTRL mientras se hace clic con el botón izquierdo sobre el test a actualizar. Después hacer clic en Aceptar.
- Hacer clic en SetUp/Salvar a disco. El ordenador emitirá dos bips rápidos.

B.3.3. Borrar un test.

Si se desea borrar un test, se deberá proceder de la siguiente manera:

- Hacer click sobre el test que se desea borrar.
- Hacer clic en SetUp/Borrar SetUp.
- Hacer clic en SetUp/Salvar a disco. El ordenador emitirá dos bips rápidos.

Bibliografía

- [1] Núñez-Peña M. I. Potenciales evocados cerebrales en el contexto de la investigación psicológica: una actualización. *Anuario de Psicología. Facultat de Psicologia. Universitat de Barcelona.*, 35(1):3–21, 2004.
- [2] JSXGraph. Jsxgraph reference. <http://jsxgraph.uni-bayreuth.de/docs/index.html>.
- [3] Michael D. Rugg and Michael G. H. Coles. *Electrophysiology of mind: event-related brain potentials and cognition*. Oxford University Press, 1996.
- [4] A. Becerra and M. Torres. *Desarrollo de aplicaciones PHP para bases de datos*. Universidad de Almería, 2007.
- [5] Goodman D., Morrison M., Novitski P., and Gustaff Rayl C. *JavaScript Bible, 7th Edition*. John Wiley and Sons, 2009.
- [6] M. Torres and S. Martínez. *Introducción al desarrollo de aplicaciones web ricas para bases de datos usando Ajax*. Universidad de Almería, 2011.
- [7] Pressman Roger S. *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill, 2002.
- [8] Sommerville I. *Ingeniería del Software. 6th Edition*. Adison Wesley, 2002.
- [9] Miraya F. Seminario taller sobre potenciales evocados auditivos. nociones técnicas. Technical report, Universidad Nacional de Rosario, 2008.
- [10] T. Adrián de Pérez. Obtencion de potenciales evocados individuales. <http://prof.usb.ve/tperez/investigacion/cancelacion/pe/PECancelac.htm>.
- [11] Wikipedia. Potenciales evocados visuales. http://es.wikipedia.org/wiki/Potenciales_Evocados_Visuales.
- [12] Gabinete Neurofisiología Otológica. Potenciales evocados auditivos y vestibulares. <http://vertigo.vertigo-neurofisiologia.com>.
- [13] O. Papazian, I. Alfonso, and R. Luzondo. Indicaciones médicas y quirúrgicas de los potenciales evocados. *Actualizaciones en Neurología Infantil*, 67(6/1):647–660, 2007.
- [14] Wikipedia. La onda p300. <http://es.wikipedia.org/wiki/P300>.

- [15] J.A. Periañez Morales and Barceló Galindo F. Electrofisiología de las funciones ejecutivas. <http://www.serviciodc.com/congreso/congress/pass/conferences/Perianez.html>.
- [16] M. J. Presentación Herrero and M. A. Martínez Benedicto. Trastorno por déficit de atención con hiperactividad y potenciales evocados cognitivos. proyecto de investigación. Technical report, Universitat Jaume I, 2008.
- [17] J.A. Portellano Pérez. Intervención global del trastorno por déficit de atención. un abordaje neuropsicológico. Technical report, Universidad Complutense de Madrid, 2000.
- [18] Programación en castellano. Jimi Marley. ¿por qué elegir php? http://www.programacion.com/articulo/por_que_elegir_php_143.
- [19] Netconcepts. Stephan Spencer. Php versus asp comparison. <http://www.netconcepts.com/php-versus-asp/>.
- [20] Bianca Valentin and Peter Wilfahrt. Lightning fast data plotting and graphics with jsxgraph at svg open, 8th international conference on scalable vector graphics, paris. Technical report, University of Bayreuth, Germany, 2010.
- [21] Wikipedia. Scalable vector graphics. http://es.wikipedia.org/wiki/Scalable_Vector_Graphics.
- [22] Bianca Valentin and Michael Gerhäuser. *Interactive SVG with JSXGraph*. JSXGraph, 2009.
- [23] Michael Gerhäuser and Bianca Valentin. Interactive svg with jsxgraph at svg open, 7th international conference on scalable vector graphics, mountain view. Technical report, University of Bayreuth, Germany, 2009.
- [24] World Wide Web Consorci W3C. Guía breve de css. <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
- [25] Psicobyte. Estructura de una hoja de estilo css. <http://www.psicobyte.com/html/css/css2.html>.
- [26] World Wide Web Consorci W3C. Specifications and drafts. <http://www.w3.org/Style/CSS/>.
- [27] Javier Eguíluz Pérez. *Introducción a AJAX*. Librosweb, 2008.
- [28] S. Martínez and M. Torres. *Introducción al Desarrollo de Aplicaciones Web Ricas para Bases de Datos usando Ajax*. Universidad de Almería, 2011.
- [29] A. Durán. Un entorno metodológico de ingeniería de requisitos para sistemas de información. Technical report, Universidad de Sevilla, 2000.
- [30] M.T. Daza and N. García-Giménez. Estudio atencional en niños con riesgo en los principales rasgos del tdah. *Revista de Neurología*, 51:244–244, 2010.

- [31] Keith H. Chiappa. *Evoked potentials in clinical medicine. 3rd Edition.* Lippincott-Raven, 2007.
- [32] Medelec. *Clinical Applications of Magneto-electrical Evoked Potentials. Application Notes (32 pages).* Surrey, England: Medelec, 1983-1986.
- [33] Weir J., Walter G. Bradley, and Robert B. Daroff. *Neurología clínica: diagnóstico y tratamiento. 4^a Edición.* Elsevier Inc, 2005.